Approaches to Hungarian Named Entity Recognition

Eszter Simon

A Thesis submitted for the degree of Doctor of Philosophy
PhD School in Psychology – Cognitive Science
Budapest University of Technology and Economics

Budapest, 2013

Supervisor:
András Kornai

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, András Kornai, for his patience and encouragement during the long years of my PhD studies. I would like to also say thanks to my boss, Tamás Váradi, who provided me the research environment and enough time for completing my dissertation.

My thanks go to my colleagues at the Research Group for Language Technology at RIL HAS and SZTAKI for all of their help and the inspiring atmosphere. Special thanks go to Gábor Recski, who helped me with proof-reading, and to Attila Zséder and Judit Ács, who helped my work on feature engineering, even in the last minutes.

Since a computational linguist always work within a research team, I have a lot of co-authors, without whom none of this would be possible. They are from different research fields from psycholinguistics to mathematics: Anna Babarczy, Ildikó Bencze, Richárd Farkas, István Fekete, Péter Halácsy, Iván Mittelholcz, Dávid Nemeskey, Péter Rebrus, András Rung, Bálint Sass, András Serény, György Szarvas, Viktor Trón, Péter Vajda, and Dániel Varga.

My first work place, where my career as a computational linguist began, was Magyar Nagylexikon Kiadó Zrt., where I had excellent colleagues, to whom I am very grateful for all their encouragement and criticism. They are: György Gyepesi, Lajos Incze, Árpád Kiss, and Zsolt Czinkos.

I wish to thank my family and my friends for their support and patience. There are no words to express how grateful I am to my Mother, without whom I think I have never reached anything in my life. And last but not least, I thank Iván, my husband, for all his never-ceasing patience and support.

Summary

Computational Linguistics (CL) is an interdisciplinary field of computer science and linguistics concerned with the computational aspects of human language faculty. Information Extraction (IE) is one of the main subtasks of CL, aiming at automatically extracting structured information from unstructured documents. It covers a wide range of subtasks from finding all the company names in a text to finding all the actors of an event. Such capabilities are increasingly important for sifting through the enormous volumes of text to find pieces of relevant information. Named Entity Recognition (NER), the task of automatic identification of selected types of Named Entities (NEs), is one of the most intensively studied tasks of IE.

The thesis presents the main issues of NER, concentrating on the Hungarian language. Since the focus of CL research has mostly been on the English language, it is also discussed.

Chapter 1 gives an overview of the thesis, enumerates my publications and my contribution to several tasks.

In Chapter 2, the key issue is how to define NEs. After studying the annotation guidelines generally used in NER, I concluded that a stronger definition is needed. For this purpose, I studied language philosophical views and the linguistic background of the theory of proper names.

In Chapter 3, I give an overview of metonymy types, and present a maximum entropy based system, which achieved the best overall results in the SemEval-2007 metonymy resolution shared task.

Chapter 4 introduces the gold standard corpora used in NER. I present a new method to create automatically NE tagged English and Hungarian corpora built from Wikipedia.

In Chapter 5, rule-based and statistical NER systems are presented, in whose development I participated. Our statistical NE tagger achieves the best overall F-measure for Hungarian.

In Chapter 6, I describe the features generally used for NER, and provide results about their power. I also study the effects of gazetteer list size on the performance of NER systems.

Contents

List of Tables

List of Abbreviations

ACE	Automatic Content Extraction
ACL	Association for Computational Linguistics
BNC	British National Corpus
CJKV	Chinese, Japanese, Korean, Vietnamese
CL	Computational Linguistics
CoNLL	Conference on Computational Natural Language Learning
CRF	Conditional Random Field
ENAMEX	Entity Expression
FN	false negative
FP	false positive
GATE	General Architecture for Text Engineering
HLT	Human Language Technology
HMM	Hidden Markov Model
IE	Information Extraction
IR	Information Retrieval
JAPE	Java Annotation Patterns Engine
LCTL	Less Commonly Taught Languages
LDC	Linguistic Data Consortium
MEMM	maximum entropy Markov model
MET	Multilingual Entity Task
MNL	Magyar Nagylexikon (Hungarian Encyclopedia)
MT	Machine Translation
MUC	Message Understanding Conference
NE	Named Entity
NER	Named Entity Recognition
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NP	noun phrase

NUMEX	Numeric Expression
OOV	out-of-vocabulary
PMW	potentially metonymic word
POS	part-of-speech
SASWP	Semantically Annotated Snapshot of the English Wikipedia
SemEval	Semantic Evaluation
SVM	Support Vector Machine
TIMEX	Time Expression
TN	true negative
TP	true positive
WSD	Word Sense Disambiguation

Chapter 1

Overview and Theses

Computational Linguistics (CL) is an interdisciplinary field of computer science and linguistics concerned with the computational aspects of human language faculty. It belongs to the cognitive sciences and overlaps with the field of artificial intelligence, a branch of computer science aiming at computational models of human cognition. The theoretical aim of CL is to build formal theories and models about the linguistic knowledge that a human needs for generating and understanding language. However, CL has an applied component as well, which is often called Human Language Technology (HLT), and is used to develop software systems designed to process or produce different forms of human language.

Information Extraction (IE) is one of the main subtasks of CL, aiming at automatically extracting structured information from unstructured or semi-structured machine-readable documents. It covers a wide range of subtasks from finding all the company names in a text to finding all the actors of an event, for example to know who killed whom, or who sold their shares to whom. Such capabilities are increasingly important for sifting through the enormous volumes of online text to find pieces of relevant information the user wants.

Named Entity Recognition (NER), the task of automatic identification of selected types of Named Entities (NEs), is one of the most intensively studied tasks of IE. Presentations of language analysis typically begin by looking words up in a dictionary and identifying them as nouns, verbs, adjectives, etc. But most texts include lots of names, and if a system cannot find them in the dictionary, it cannot identify them, making it hard to produce a linguistic analysis of the text. Thus, NER is of key importance in many Natural Language Processing (NLP) tasks, such as Information Retrieval (IR) or Machine Translation (MT).

1.1 The Definition of Named Entities

The NER task, which is often called as Named Entity Recognition and Classification in the literature, has two substeps: first, locating the NEs in unstructured texts, and second, classifying them into pre-defined categories.

A key issue is how to define NEs. This issue interconnects with the issue of selection of classes and the annotation schemes applied in the field of NER. The NER task was introduced with the 6th Message Understanding Conference (MUC) in 1995 [Grishman and Sundheim, 1996], consisting of three subtasks: recognizing entity names, temporal and numerical expressions. Although there is a general agreement in the NER community about the inclusion of temporal expressions and some numerical expressions, the most studied types are names of persons, locations and organizations. The fourth type, called `Miscellaneous`, was introduced in the NER tasks of the Conference on Computational Natural Language Learning (CoNLL) in 2002 [Tjong Kim Sang, 2002] and 2003 [Tjong Kim Sang and De Meulder, 2003], and includes proper names falling outside the three classic types. Since then, MUC and CoNLL datasets and annotation schemes have been the major standards applied in the field of NER.

The annotation guidelines of these shared tasks are based on examples and counterexamples of what to annotate as a NE, rather than an exact, theoretically well-founded definition of NEs. The next description is from the MUC-7 Named Entity Task Definition [Chinchor, 1998a]:

> "This subtask is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via the TYPE attribute as follows:
> ORGANIZATION: named corporate, governmental, or other organizational entity
> PERSON: named person or family
> LOCATION: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)"

Besides this description negative examples (non-entities) are also provided. For annotating texts with NE labels, this kind of definition is not really helpful. In addition, the annotation guidelines mentioned above contain instructions only for English entities and non-entities. But in other languages, e.g. in Hungarian, there are concepts which would be annotated as NEs according to these guidelines, but they are not proper names,

and thus are not considered as NEs. During the work of writing annotation guidelines for Hungarian [Simon et al., 2006] based on the widely used guidelines, their weak points became evident. From these experiences we conclude that a stronger definition is needed for annotation of NEs.

For this purpose, we studied Kripke's theory [Kripke, 2000] about the proper names as rigid designators. Kripke broke up with Frege's [Frege, 2000] and Russell's [Russell, 2000] description theory of proper names. In Chapter 2, we give an overview of the philosophic and linguistic background of the theory of proper names. After discussing the theoretical background, we try to map our findings to the NER task.

Thesis 1 *After investigating several theories of proper names, we can conclude that for getting a usable definition of NEs, the classic Aristotelian view on classification, which states that there must be a differentia specifica which allows something to be the member of a group, and excludes others, is not applicable. For our purposes, the prototype theory seems more plausible, where proper names form a continuum ranging from prototypical (person and place names) to non-prototypical categories (product and language names). Finally, the goal of the NER application will further restrict the range of linguistic units to be taken into account.*

The author's contribution. The author participated in the work which aimed at building a large, heterogeneous, manually NE annotated Hungarian corpus called the HunNer corpus. The author prepared the annotation scheme, and wrote the guidelines. For reasons outside the author's control, the HunNer corpus is still not entirely complete, but the guidelines have been used for other projects, e.g. for building the Criminal NE corpus[1]. These results are partly described in [Simon, 2008] and [Simon et al., 2006] and in the annotation guidelines, which is accessible on the web through the URL `http://krusovice.mokk.bme.hu/~eszter/utmutato.pdf`.

1.2 Handling Metonymic Named Entities

In metonymy, the name of one thing is substituted for that of another related to it [Lakoff and Johnson, 1980]. Besides common nouns, many proper names are widely used metonymically, as it can be seen in Examples 1.1 and 1.2. (Examples of metonymic NEs are not intuitively created by us, but they are accurate linguistic samples from the datasets

[1] http://www.inf.u-szeged.hu/rgai/nlp?lang=en&page=corpus_ne

provided by the organizers of SemEval-2007 metonymy resolution shared task [Markert and Nissim, 2007b], various articles and the web. In examples, throughout the dissertation, the relevant parts are italicized, or, if tags are important, they are in square brackets, with the tags in subscript.)

(1.1) Denise drank *the bottle*.

(1.2) Ted played *Bach*.

None of the two sentences is literally true. In Example 1.1, Denise did not drink the bottle made of plastic or glass, but the liquid in the bottle. In Example 1.2, Ted did not play the person whose name is Bach, but music composed by Bach [Fass, 1988].

This type of reference shift is very systematic, in that it can occur with any person name, as long as the discourse participants are aware of that he/she is an artist, and they can associate an artwork with him/her. Linguistic studies (e.g. Lakoff and Johnson [1980]; Fass [1988]) postulate conventional metonymies that operate on semantic classes (here: person, location, and organization names). A few examples of such conventional metonymies follow (the standard name of metonymies are indicated with small capitals after the example sentences, in parentheses):

(1.3) *Spain* won its third straight major soccer title Sunday. (PLACE FOR PEOPLE)

(1.4) The broadcast covered *Vietnam*. (PLACE FOR EVENT)

(1.5) *Apple* announced new iPads and Mac computers. (ORGANIZATION FOR MEMBERS)

(1.6) It was the largest *Fiat* anyone had ever seen. (ORGANIZATION FOR PRODUCT)

Besides such regular shifts, metonymies can also be created on the fly: in Example 1.7, 'seat 19' refers to the person occupying seat 19. Markert and Nissim [2007a] call such occurrences unconventional metonymies.

(1.7) Ask *seat 19* whether he wants to swap.

Apart from being regular and productive, metonymic usage of NEs is frequent in natural language. State-of-the-art NER sytems usually do not distinguish between literal and metonymic usage of names, even though it would be helpful for most applications. Resolving metonymic usage

of proper names would therefore directly benefit NER and indirectly all NLP tasks that require NER. The importance of resolving metonymies has been shown for a variety of NLP tasks, e.g. MT [Kamei and Wakao, 1992], question answering [Stallard, 1993], and anaphora resolution [Harabagiu, 1998; Markert and Hahn, 2002].

Distinguishing literal and metonymic usage, then identifying the intended referent can be seen as a classification task. Markert and Nissim [2002] postulate the metonymy resolution task as comparable to the Word Sense Disambiguation (WSD) task, so that metonymies can be recognized automatically with similar methods. On this assumption, Markert and Nissim [2007b] organized a shared task of the 2007 evaluation forum of the Semantic Evaluation series (SemEval-2007), which aimed at recognition and categorization literal, mixed, and metonymic usage of location and organization names. We built a maximum entropy based system [Farkas et al., 2007], which achieved the best overall results in the competition. In Chapter 3, we give an overview of conventional and unconventional metonymies, and present the system description.

Thesis 2 *Since conceptual mappings between the related referents of metonymic words are not linked to particular linguistic forms, recognizing metonymic NEs is quite difficult. However, using some surface and syntactic information, and applying several semantic generalization methods lead to improvement in resolving metonymies. We present a supervised system, which achieved the best overall results in the SemEval-2007 metonymy resolution task. As our results show, the main dividing line does not lie between conventional and unconventional metonymies, rather between literal and metonymic usage.*

The author's contribution. Building the metonymy resolution system was a joint effort with the co-authors, namely Richárd Farkas, György Szarvas, and Dániel Varga. The author is responsible for investigating the related work, and providing the theoretical background. In addition, the author is responsible for some semantic generalization features, in particular for using Levin's verb classes and collecting the trigger words. The author also participated in feature engineering to find out whether each feature has the requisite discriminative power, the evaluation of results, and the drawing of conclusions. These findings are described in Farkas et al. [2007] and partly in Simon [2008].

1.3 Gold and Silver Standard Corpora for Named Entity Recognition

The supervised statistical approach requires a large amount of texts to boost performance quality. Such a large and structured set of texts is called a *corpus*. Corpora can be classified according to different criteria: they can be general or domain-specific, monolingual or multilingual, tagged or untagged. To be a gold standard corpus, a dataset has to meet several requirements, for example to be exhaustive or aiming for representativeness; to be large enough for training and testing supervised systems on it; and to contain accurate linguistic annotation added by hand.

The gold standard corpora in the field of NER are highly domain-specific, containing mostly newswire, and are restricted in size. Researchers attempting to merge these datasets to get a bigger training corpus are faced with the problem of combining different tagsets and annotation schemes. Manually annotating large amounts of text with linguistic information is a time-consuming, highly skilled and delicate job, but large, accurately annotated corpora are essential for building robust supervised machine learning NER systems. Therefore, reducing the annotation cost is a key challenge.

There are more ways to reach this goal. One approach is to use semi-supervised or unsupervised methods, which do not require large amount of labelled data. Another approach is to generate the resources automatically, or at least applying NLP tools that are accurate enough to allow automatic annotation. Yet another approach is to use collaborative annotation and/or collaboratively constructed resources, such as Wikipedia or DBpedia. Here we present a method which combines these approaches by automatically generating freely available NE tagged corpora from Wikipedia.

An automatically generated or silver standard corpus provides an alternative solution which is intended to serve as an approximation to a gold standard corpus. Such corpora are very useful for improving NER in several ways.

In Chapter 4, first, we give an overview of corpus building in general (Section 4.1). Section 4.2 introduces the gold standard corpora used in NER. In Section 4.3, we present our method to create automatically NE tagged English and Hungarian corpora built from Wikipedia.

Thesis 3 *We present a new method with which we can get closer to one of the main goals of current NER research, i.e. reducing the annotation labour of corpus building. We built automatically generated NE tagged corpora from Wikipedia for*

English and Hungarian. The one presented here is the first automatically NE annotated corpus for Hungarian which is freely available. As for English, there are no such automatically built corpora freely available, except for the Semantically Annotated Snapshot of the English Wikipedia, but their method cannot be applied for less resourced languages. As our method is mainly language-independent, it can be applied for other Wikipedia languages as well.

Thesis 4 *We showed that automatically generated silver standard corpora are very useful for improving NER in several ways: (a) for less resourced languages, they can serve as training corpora in lieu of gold standard datasets; (b) they can serve as supplementary or independent training sets for domains differing from newswire; (c) they can be sources of huge entity lists, and (d) feature extraction.*

The author's contribution. The author participated in several corpus building projects.

Within the Hungarian Diachronic Generative Syntax project, the author is responsible for building a corpus which contains all text sources from the Old Hungarian period and a balanced selection from the Middle Hungarian period. The corpus is available via an online search engine: `http://rmk.nytud.hu/`. Related publications: Simon et al. [2011]; Simon and Sass [2012]; Oravecz et al. [2010].

Within the ABSTRACT project, which was a multi-site, EU funded research project that investigated how abstract linguistic concepts are learned and represented by the human mind, the author is responsible for building a corpus containing annotation of metaphorical expressions. Several methods were investigated for automatic identification of metaphors. The findings and the corpus itself are published in Babarczy et al. [2010a,b] and Babarczy and Simon [2012].

Within the HunNer corpus project, the author is responsible for preparing the annotation scheme and writing the guidelines. The corpus is described in Simon et al. [2006].

Building the silver standard corpora for English and Hungarian was a joint effort with the co-author, Dávid Nemeskey. The author is responsible for investigating the related work, and providing the linguistic background. In addition, the author contributed to the construction of mapping between DBpedia ontology classes and gold standard tagsets, handling several problematic cases of NE labelling, and analysing and evaluating the error types of our method. Experiments for evaluating the newly generated datasets are the author's work. The method and the corpora themselves are published in Simon and Nemeskey [2012] and Nemeskey and Simon [2012].

1.4 Approaches to Named Entity Recognition

The NER task, similarly to other NLP tasks, can be approached in two main ways: by applying hand-crafted rules, or by statistical machine learning techniques. This dichotomy is typical in the entire field of NLP, which dated back to end of the 1950s, when Chomsky published his influential review of Skinner's *Verbal Behavior* [Chomsky, 1959]. Finite state and probabilistic models, which were widely used before, had lost popularity in this period, and NLP split very cleanly into two paradigms, the theory-oriented or rule-based, and the data-driven or stochastic paradigms. In the early 1990s, the success of statistical methods in speech spread to other areas of NLP. This period has been called as the "return of empiricism". Due to the philosophical background of the paradigms they have also been called rationalist and empiricist approaches. Section 5.1 gives an overview of the philosophical background and the history of the two camps, until recent years, when the field comes together, and researchers try to build hybrid systems reaping the benefits of both approaches.

A rule-based NER application requires patterns which describe the internal structure of names and context-sensitive rules which give clues for classification. In Section 5.2, we give an enumeration of several kinds of internal end external evidence of NER, and describe a rule-based system using such patterns to extract NEs from Hungarian encyclopedic texts. We point to the disadvantages of rule-based systems, and conclude that applying machine learning algorithms is more useful for NER.

Statistical machine learning algorithms can be classified according to the type of input data they need. Unsupervised learning means that we do not have linguistically annotated data, thus the challenge is finding hidden structure in unlabelled data. Semi-supervised learning combines both labelled and unlabelled examples to generate an appropriate classifier. NLP tasks can also be solved by using labelled corpora and supervised learning methods that induce rules by discovering patterns in the manually annotated source text.

For building a supervised NER system, first we need a manually annotated gold standard corpus, which contains linguistic information. Typically, the algorithm itself learns its parameters from the corpus, and the evaluation of the system is through comparing its output to an other part of the corpus. So the corpus is divided into two parts: a training and a test set. When building a supervised learning system, a major step is feature extraction, that is collecting information from the data that can be relevant for the task. These features are the input of the learning algorithm that builds a model based on the regularities found in the data. After that the

test set is tagged with the most probable labels, then they are compared to the gold standard labels. The evaluation means here to quantify the similarity between the two labellings. The whole process from training to evaluating a supervised NER system is described in details in Subsection 5.3.1.

For major languages, hundreds of papers were published on NER systems based on several supervised machine learning techniques. There are not too many language-dependent components of these, yet for Hungarian, we are aware only of one quantitative study of a NER system which is based on machine learning methods [Szarvas et al., 2006b]. Our statistical NE tagger, the `hunner` system overperforms that system, achieving the best F-measure for Hungarian. In Subsection 5.3.2, we give a detailed system description.

Thesis 5 *The NER task, similarly to other NLP tasks, can be resolved by applying hand-crafted rules or machine learning techniques. We present a rule-based system developed for recognizing NEs in Hungarian encyclopedic texts and a supervised machine learning NER system which achieved the best performance for Hungarian. As our results show, applying statistical algorithms results in a more robust system and in higher performance on Hungarian NER.*

The author's contribution. The author contributed to several works concerned with rationalist and empiricist approaches to language acquisition as well as to NLP tasks.

The author participated in the 'Analogical generalisation processes in language acquisition' project, which had the aim of modelling the mechanisms of child language acquisition, specifically the process of learning argument structures from the input available to young children. We applied several statistical models for the automatic acquisition of subcategorization frames, and we concluded that data frequency and the size of the input corpus are important factors in both psycholinguistics and machine learning. These findings are published in Serény et al. [2009]; Babarczy et al. [2009] and Simon et al. [2010].

Within the Hungarian Diachronic Generative Syntax project, the author participated in the development of a semi-automatic text normalization system applied for Old Hungarian texts. Most of the work on text normalization of historical documents is centered around a manually crafted set of correspondence rules. In contrast, we used the noisy channel paradigm to build an automatic normalization system. The human labour has been shifted to building training data for the transliteration model, for which the author is responsible. By the means of automatic normalization,

the manual annotation process can be reduced to a selection of the right solution from the list of candidates provided by the system. The methodology and the results are presented in Oravecz et al. [2009, 2010].

The rule-based system developed for recognizing NEs in a Hungarian encyclopedia, Magyar Nagylexikon, remained unpublished, because it was treated with confidentiality. The system development was a joint effort with the colleagues, György Gyepesi, Lajos Incze, Zsolt Czinkos and Árpád Kiss. The author is responsible for creating the NE affixing rules and the transcribing rules for 20 languages, constructing and manually checking the gazetteer lists, and writing regular expression patterns providing information about the NEs' internal and external evidence.

The development of the original `hunner` system was a joint effort with the co-author, Dániel Varga. The author is responsible for feature engineering, data collection and evaluation. The author did not participate in the system's reimplementation, but is responsible for implementing and testing new features and collecting new gazetteers. The original system is published in Varga and Simon [2006, 2007].

1.5 Feature Engineering

Features are descriptors or characteristic attributes of datapoints in a text. In token-based classification tasks of NLP, feature vectors are assigned to every token, where the feature vector contains one or more features. Generally, Boolean- or string-valued features are applied in NER. For example, if a word is capitalized, it gets an `iscap=1` feature. Feature vector representation is a kind of abstraction over text. The task of the machine learning algorithm is then to find regularities in this large amount of information that are relevant for NER.

Defining features for a supervised system is a manual work, similarly to coding patterns for a rule-based system. In the statistical methodology, however, the linguist does not tell anything about the power of the features, but it is found out from the corpus. The human cognition tends to realize only salient phenomena, thus declare features as important ones which are then found out not to be important based on corpus data, and vice versa. For this reason, the power of every feature has to be measured on real data before inclusion into the system. This is called feature engineering.

To measure the strength of features, we virtually built NER systems for Hungarian and English by adding new features to them one by one. For this purpose, we used the reimplemented version of the `hunner` system.

In Chapter 6, we describe the features generally used for NER, and provide results about their power. We organize the features along the dimension of what kind of properties they provide: surface properties, digit patterns, morphological or syntactic information, or gazetteer list inclusion. As for the last kind of features, we also study the effects of gazetteer list size on the performance of NER systems.

Thesis 6 *We present a way of feature engineering in which the features most often used in NER are measured for getting the knowledge about their discriminative power. We conclude that for a supervised NER system the string-valued features related to the character makeup of words are the strongest features. Quite counterintuitively, features indicating casing information and sentence starting position do not improve the performance. Features based on external language processing tools such as morphological analysers and chunkers are also not necessary for finding NEs in texts.*

Thesis 7 *We compare the performance of a maximum entropy NER system under widely different entity list size conditions, ranging from a couple of hundred to several million entries, and conclude that for statistical NER systems entity list size has only a very moderate impact. If large entity lists are available, we can use them, but their lack does not cause invincible difficulties in the development of NER systems.*

The author's contribution. Defining most features presented in Chapter 6, measuring and evaluating them is the author's own work. Pre-processing of the Hungarian and English data and enriching them with linguistic information so serving as an appropriate input corpus for NER is also the author's own work. (Except for mapping the chunk tags of the Szeged Treebank to the Szeged NER corpus, which is the work of Attila Zséder and Judit Ács.) Collecting and designing the gazetteers used in the experiments is also the author's own work.

The author contributed to the development of the Hungarian `morphdb`, a lexical database and morphological grammar, which was used for the morphological analysis of the input corpora used for NER. It is published in Trón et al. [2005b, 2006a,b].

The author contributed to the work of designing a system for recognizing metaphorical expressions by the means of different kinds of lists. The author is responsible for designing the lists, developing the software environment, and building the corpora on which the methods were evaluated. One of the important findings of this work is that using accurately compiled lists by hand is the most successful method for recognizing the

relevant elements in a text. These findings are published in Babarczy et al. [2010a,b] and Babarczy and Simon [2012].

The author contributed to several works on feature engineering of state-of-the-art NER systems: to building a system for recognizing metonymic NEs in English texts (cf. Chapter 3) and to building the original `hunner` system (cf. Chapter 5). In both of them, the author is responsible for defining new features and measuring their strength. These findings are published in Farkas et al. [2007] and Varga and Simon [2006, 2007].

Chapter 2

The Definition of Named Entities

The major standard guidelines applied in the field of NER do not give an exact definition of NEs, but rather list examples and counterexamples. The only common statement they make is that NEs have unique reference. For getting a usable definition of NEs, we investigate the approach taken in the philosophy of language and linguistics, and we map our findings to the NER task. We do not wish to give a complete description of the theory and typology of proper names, but to find a plausible way to define linguistic units relevant to the NER task.

The chapter is structured as follows. In Section 2.1, we give an overview of the annotation schemes applied in the field of NER. Section 2.2 describes the philosophical approach, and Section 2.3 gives the linguistic background of the theory of proper names. Section 2.4 concludes the chapter with the most important findings about mapping the theory of proper names to the NER task.

2.1 Annotation Schemes

The first major event dedicated to the NER task was the *MUC-6* in 1995. As the organizers write in their survey about the history of MUCs [Grishman and Sundheim, 1996], these conferences were rather similar to shared tasks, because participants were required to submit their results to attend the conference. Prior MUCs focused on IE tasks; MUC-6 was the first including the NER task, which consisted of three subtasks [Sundheim, 1995]:

- entity names (ENAMEX): organizations, persons, locations;
- temporal expressions (TIMEX): dates, times;

- number expressions (NUMEX): monetary values, percentages.

The annotation guidelines define NEs as "unique identifiers" of entities, and give an enormous list of what to annotate as NEs. However, the best support for annotators is the restriction about what not to annotate: "names that do not identify a single, unique entity".

As for the temporal expressions, the guidelines distinguish between absolute and relative time expressions. To be considered absolute, the expression must indicate a specific segment of time, e.g.

(2.1) *twelve o'clock noon*

(2.2) *January 1979*

A relative time expression indicates a date relative to the date of the document, or a portion of a temporal unit relative to the given temporal unit, e.g.

(2.3) *last night*

(2.4) *yesterday evening*

In MUC-6, only absolute time expressions were to be annotated.

The numeric expressions subsume monetary and percentage values. Modifiers that indicate the approximate value of a number are to be excluded from annotation, e.g.

(2.5) about *5%*

(2.6) over *$90,000*

The modified version of MUC-6 guidelines were used for *MUC-7* NER task in 1998 [Chinchor, 1998a]. The most notable change was that relative time expressions became taggable. The MUC-7 guidelines became one of the most widely used standards in the field of NER. They were used with slight modifications for the Multilingual Entity Tasks (MET-1 and 2) [Merchant et al., 1996] and for the Hub-4 Broadcast News Evaluation [Miller et al., 1999] in 1999.

According to the MUC guidelines embedded NEs can also be annotated, e.g.

(2.7) The [morning after the [July 17]$_{DATE}$ disaster]$_{TIME}$

The *CoNLL* conference is the yearly meeting of the Special Interest Group on Natural Language Learning (SIGNLL) of the Association for Computational Linguistics (ACL). Shared tasks organized in 2002 and 2003 were concerned with language-independent NER [Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003]. Annotation guidelines were based on the NER task definition of the MITRE Corporation[1] and the Science Applications International Corporation (SAIC) [Chinchor et al., 1999], which are slightly modified versions of the MUC guidelines. A new type, `Measure`, was introduced for NUMEX elements, e.g.

(2.8) *23 degrees Celsius*

In contrast to the MUC guidelines, instructions are given regarding certain kinds of metonymic proper names (see Chapter 3 for details), decomposable and non-decomposable names, and miscellaneous non-taggables. The latter constitute a new category, `Miscellaneous`, which includes names falling outside the classic ENAMEX, e.g. compounds that are made up of locations, organizations, etc., adjectives and other words derived from a NE, religions, political ideologies, nationalities, or languages.

As part of the *Automatic Content Extraction* (ACE) program (a series of IE technology evaluations from 1999 organized by the National Institute of Standards and Technology (NIST)), new NE types were introduced in addition to the classic ENAMEX categories: `Facility`, `Geo-Political Entity`, `Vehicle` and `Weapon`. The category `Facility` subsumes artifacts falling under the domains of architecture and civil engineering. `Geo-Political Entities` are composite entities comprised of a population, a government, a physical location, and a nation (or province, state, county, city, etc.). The seven main types are divided into dozens of subtypes and hundreds of classes [ACE, 2008]. The ACE program is concerned with automatic extraction of content, including not only NEs but also their relationships to each other and events concerning them. For the purposes of this more complex task, all references to entities are annotated: names, common nouns, noun phrases, and pronouns. In this regard, ACE is exceptional in the race of NER standards, where common nouns and pronouns are not to be annotated.

The Linguistic Data Consortium (LDC) has developed annotation guidelines for NEs and time expressions within the *Less Commonly Taught Languages* (LCTL) project. In contrast to the ones mentioned above, these guidelines give an exact definition of NEs [Linguistic Data Consortium LCTL Team, 2006]: "An entity is some object in the world – for instance,

[1] `http://www.mitre.org/`

a place or a person. A named entity is a phrase that uniquely refers to that object by its proper name, acronym, nickname or abbreviation." Besides the classical name categories (`PER`, `ORG`, `LOC`), they also annotate `Titles`, which are separated from the person's name, e.g.

(2.9) said [GlobalCorp]$_{ORG}$ [Vice President]$_{TTL}$ [John Smith]$_{PER}$

The LCTL annotation guidelines are the first concerned with meaning and compositionality of NEs: "The meaning of the parts of names are not typically part of the meaning of the name (i.e. names are not *compositional*) and, therefore, names cannot be broken down into smaller parts for annotation." Thus, a NE is treated as an indivisible syntactic unit that cannot be interrupted by an outside element.

In addition to the classical ENAMEX, TIMEX and NUMEX categories, there are a wide range of other, marginal types of NEs, which are relevant for particular tasks, e.g. extracting chemical and drug names from chemistry articles [Narayanaswamy et al., 2003]; names of proteins, species, and genes from biology articles [Rindfleish et al., 2000]; or project names, email addresses and phone numbers from websites [Zhu et al., 2005].

Summary. Early works define the NER problem as the recognition of proper names in general. Names of persons, locations and organizations have been studied the most. Besides these classical categories, there is a general agreement in the NER community about the inclusion of temporal expressions and some numerical expressions, such as amounts of money and other types of units. The main categories can be divided into fine-grained subtypes and classes, and marginal types are sometimes included for specific tasks. Annotation guidelines usually do not go further in defining NEs than saying that they are "unique identifiers" or that they "uniquely refer" to an entity. Only one of the guidelines mentions the meaning and compositionality of NEs: it postulates NEs as indivisible units, although earlier guidelines allow embedded NEs.

2.2 Language Philosophical Views: from Mill to Kripke

"A proper name is a word that answers the purpose of showing what thing it is that we are talking about, but not of telling anything about it", writes *John Stuart Mill* in his 1843 *A Sytem of Logic* [Mill, 2002]. According to him, the semantic contribution of a name is its referent and only its referent. One of his examples illustrating this statement is the name of the town

Dartmouth. The town was probably named after its localization, because it lies at the mouth of the river Dart. But if the river had changed its course, so that the town no longer lay at the mouth of the Dart, one could still use the name 'Dartmouth' to refer to the same place as before. Thus, it is not part of the meaning of the name 'Dartmouth' that the town so named lies at the mouth of the Dart.

Gottlob Frege's puzzle of the Morning Star and the Evening Star challenges the Millian conception of names. In his famous work *Über Sinn und Bedeutung* [Frege, 2000], he distinguishes between *sense* (Sinn) and *reference* (Bedeutung). Without the distinction between sense and reference, the following sentences would be equal:

(2.10) The Morning Star is the Evening Star.

(2.11) The Morning Star is the Morning Star.

Both names have the same reference (Venus), so they should be interchangeable. However, since the thought expressed by Example 2.10 is distinct from the thought expressed by Example 2.11, the senses of the two names are different. While Example 2.11 seems to be an empty tautology, Example 2.10 can be an informative statement, even a scientific discovery. If somebody did not know that the Evening Star is the Morning Star, he/she could think that Example 2.11 is true, while Example 2.10 is false.

To solve the puzzle, without resorting to a two-tiered semantic theory, *Bertrand Russell* used the description theory. The *description theory of names* states that each name has the semantic value of some definite description [Cumming, 2012]. For example, 'Aristotle' might have the semantic value of 'the teacher of Alexander the Great'. 'The Morning Star' and 'the Evening Star' might correspond in semantic value to different definite descriptions, and would make different semantic contributions to the sentences in which they occur.

Frege and Russell both argue that Mill was wrong: a proper name is a definite description abbreviated or disguised, and such a description gives the sense of the name. According to Frege, a description may be used synonymously with a name, or it may be used to fix its reference. *Saul Kripke* concurred only partially with Frege's theory. Description fixes reference, but the name denoting that object is then used to refer to that object, even if referring to counterfactual situations where the object does not have the properties in question, writes Kripke in *Naming and Necessity* [Kripke, 2000]. One of Kripke's examples is Gödel and the proof of incompleteness of arithmetic. If it turned out that Gödel was not the man who

proved the incompleteness of arithmetic, Gödel would not be called 'the man who proved the incompleteness of arithmetic', but he would still be called 'Gödel'. Thus, names are not equal to definite descriptions.

Kripke postulates proper names as *rigid designator*s. Something is a rigid designator if it designates the same object in every possible world. The concept of a possible world (or counterfactual situation) is used in modal semantics, where the sentence 'Frank might have been a revolutionist' is interpreted as a quantification over possible worlds. Kripke suggests an intuitive test to find out what is a rigid designator. An updated example: 'the President of the US in 2012' designates a certain man, Obama; but someone else (e.g. Romney) may have been the President in 2012, and Obama might not have; so this designator is not rigid. When talking about what would happen to Obama in a certain counterfactual situation, we are talking about what would happen to *him*. So 'Obama' is a rigid designator.

In the case of proper names, reference can be fixed in various ways. In the case of initial baptism it is typically fixed by ostension or description. Otherwise, the reference is usually determined by a chain, passing the name from link to link. In general, the reference depends not just on what we think, but on other people in the community, the history of how knowledge of the name has spread. It is by following such a history that one gets to the reference.

Kripke argues that proper names are not the only kinds of rigid designators: species names, such as *tiger*, or mass terms, such as *gold*, certain terms for natural phenomena, such as *heat*, and measurement units, such as *one meter* are also examples. There is a difference between the phrase 'one meter' and the phrase 'the length of the metre bar at t_0'. The first phrase is meant to designate rigidly a certain length in all possible worlds, which in the actual world happens to be the length of the metre bar at t_0. On the other hand, 'the length of the metre bar at t_0' does not designate anything rigidly.

Summary. Kripke goes back to the Millian theory of names, and at the same time breaks up with Frege's theory, when he writes that proper names do not have sense, only reference. He declares that a proper name is a rigid designator, which designates the same object in every possible world. Through examples he proves that definite descriptions are not synonymous with names, but they can still fix a referent. In the case of proper names, the reference can be fixed in an initial baptism, after which the name spreads in the community by a chain, from link to link. In Kripke's theory, species names, mass terms, natural phenomena and measurement units are also rigid designators.

2.3 The Linguistic Approach

Besides the theory of rigid designators, another concept used in the literature to define NEs is that of unique reference. In Subsection 2.3.1, we clear the meaning of the phrase 'unique reference', which seems to be used non-systematically in NER guidelines. Unique reference can act as the separator line between proper names and common nouns. There are however certain *linguistic properties* by which we can make a stronger distinction, as described in Subsection 2.3.2. The main feature distinguishing between them is the issue of compositionality, which is discussed in Subsection 2.3.3. Finally, we sum up our findings about the linguistic background of proper names in Subsection 2.3.4.

2.3.1 Unique Reference

In the MUC guidelines [Chinchor, 1998a], the definition of what to annotate as NEs is as follows: "proper names, acronyms, and perhaps miscellaneous other unique identifiers", and what not to annotate as NEs: "artifacts, other products, and plural names that do not identify a single, unique entity". In the LCTL guidelines we find this definition: "a NE is a phrase that uniquely refers to an object by its proper name, acronym, nickname or abbreviation" [Linguistic Data Consortium LCTL Team, 2006].

Let's take these definitions one by one. In the first case, the phrase 'unique identifiers' is coordinated with 'proper names' and 'acronyms', and 'unique' is an attributive adjective modifying the noun 'identifiers'. So 'unique' means here that the identifier is unique, similarly to proper names and acronyms. In the second case, however, it is the entity a linguistic unit refers to that must be unique in order for the unit to qualify as a NE. In the LCTL guidelines, the phrase 'uniquely refers' means something similar as in the first case, it is therefore the referring linguistic unit that must be unique, not the entity in the world to which it refers.

Here, and several other places in the literature, the difference between the concepts of referring act and reference seems to be blurred. When trying to determine what is unique, we find that in most grammar books the names and the entities they refer to are not clearly distinguished. But it does matter whether we are talking about Charlie or about the name 'Charlie'. To prevent such an ambiguity, we always indicate the metalinguistic usage by single quotation marks.

By investigating various definition of proper names, we can conclude that names refer to a unique entity (e.g. *London*), so names have unique reference [Quirk and Greenbaum, 1980], in contrast to common nouns, which

refer to a class of entities (e.g. *cities*), or non-unique instances of a certain class (e.g. *city*). However, we can refer to and even identify an entity by means of common nouns. The difference is that proper names, even standing by themselves, always identify entities, while a common noun can do so only in such cases when it constitutes a noun phrase with other linguistic units. Common nouns may stand with a possessive determiner (e.g. *my car*), or with a demonstrative (e.g. *this car*), or can be a part of a description (e.g. *the car that I saw yesterday*).

Many proper names share the feature of having only one possible reference, but a wide range of them refer to more than one object in the world. For example, 'Washington' can refer to thousands of people who have 'Washington' as their surname or given name, a US state, the capital of the US, cities and other places throughout America and the UK, roads, lakes, mountains, educational organizations, and so forth. These kind of proper names are referentially multivalent [Anderson, 2007], but each of the references is still unique.

Some proper names occur in plural form, optionally or exclusively. In the latter case, the plural suffix is an inherent part of the name. These are the so called *pluralia tantum* (e.g. *Carpathians, Pleiades*). According to their surface form, it might seem that they can be broken down into smaller pieces, but the Carpathians do not consist of $carpathian_1$, $carpathian_2$, ..., $carpathian_n$, just as the Pleiades do not consist of pleiades. These names refer to groups of entities considered unique.

Names of brands, artifacts, and other products can be optionally used in plural form. For example, 'Volvo' is a proper name referring to a unique company. But if we put it in a sentence, like 'He likes Volvos', it will refer to particular vehicles. This is a kind of metonymy, with the company name used to refer to a product of this company (see Chapter 3 for more details). Proper names in plural form can also be used in other kinds of figures of speech, for example in metaphors. In the phrase 'a few would-be Napoleons', some characteristics of the emperor are associated with men to which the word 'Napoleons' refers. In these cases, proper names act like common nouns, i.e. they have no unique reference.

Additionally, there are a quite large number of linguistic units which are on the border between proper names and common nouns, because it is difficult to determine whether their reference is unique. Typically, they are used as proper names in some languages, but as common nouns in other ones. The difficulty of classification is usually mirrorred even in the spelling rules. For example, in the case of events (*World War II, Olympic Games* in English; *2. világháború, olimpiai játékok* in Hungarian; *Segunda Guerra Mundial, Juegos Olímpicos* in Spanish; *Seconde Guerre mondiale,*

Jeux olympiques in French), expressions for days of the week and months of the year (*Monday, August* in English; *hétfő, augusztus* in Hungarian; *lunes, agosto* in Spanish; *lundi, août* in French), expressions for languages, nationalities, religions and political ideologies (*Hungarian, Catholic, Marxist* in English; *magyar, katolikus, marxista* in Hungarian; *húngaro, católica, marxista* in Spanish; *hongrois, catholique, marxiste* in French), etc. Categories vary across languages, so there seems to be no language-independent, general rule for classifying proper names.

2.3.2 Distinction between Proper Names and Common Noun Phrases

As mentioned above, proper nouns are distinguished from common nouns on the basis of the uniqueness of their reference. However, we can make a stronger distinction based on other linguistic properties.

First, we have to clarify the disctinction between proper nouns and proper names made by current works in linguistics (e.g. [Anderson, 2007; Huddleston and Pullum, 2002]). Since the term 'noun' is used for a class of single words, only single-word proper names are proper nouns: 'Ivan' is both a proper noun and a proper name, but 'Ivan the Terrible' is a proper name that is not a proper noun. From this distinction follows that proper names cannot be compared to a single common noun, but to a noun phrase headed by a common noun. A proper noun by itself constitutes a noun phrase, while common nouns need other elements. In Subsection 2.3.1, we give a few examples. In the subsequent analysis, proper names and common noun phrases are juxtaposed.

Distinction between proper nouns and common nouns is commonly made with reference to *semantic properties*. One of them is the classic approach: entities described by a common noun, e.g. 'horse', are bound together by some resemblances, which can be summed up in the abstract notion of 'horsiness' or 'horsehood' [Gardiner, 1957]. A proper name, on the contrary, is a distinctive badge: there is no corresponding resemblance among the Charlies that could be summed up as 'Charlieness' or 'Charliehood'. Thus, we can say that common nouns realize abstraction, while proper names make distinction. However, Katz [1972] argues that the meaninglessness of names means that one cannot establish a semantic distinction between proper names and common noun phrases. The latter are compositional, because their meaning is determined by their structure and the meanings of their constituents [Gendler Szabó, 2008], while proper names "allow no analysis and consequently no interpretation of

their elements", quoting Saussure [1959]. Thus, proper names are arbitrary linguistic units, and are therefore not compositional. (See 2.3.3 for more details.)

Moving on to *syntax*, common noun phrases are compositional, i.e. they can be divided into smaller units, while proper names are indivisible syntactic units. This is confirmed by the fact that proper names cannot be modified internally, as can be seen in these examples:

(2.12) *beautiful King's College*

(2.13) **King's beautiful College*

(2.14) *my son's college*

(2.15) *my son's beautiful college*

Further evidence is that in Hungarian and other highly agglutinative languages, the inflection always goes to the end of the proper name constituting a noun phrase. Example 2.16 presents the inflection of a proper name (here: a title), while Example 2.17 shows its common noun phrase counterpart (consider the second determiner in the latter):

(2.16) *Láttam az Egerek és embereket.* 'I saw (Of Mice and Men).ACC'

(2.17) *Láttam az egereket és az embereket.* 'I saw the mice.ACC and the men.ACC'

From the perspective of *morphology*, proper names must always be sacred, which means that the original form of a proper name must be reconstructible from the inflected form [Deme, 1956]. This requirement is mirrorred even in the current spelling rules in Hungarian: e.g. *Papp-pal* 'with Papp', *Hermann-nak* 'to Hermann'. Some proper names in Hungarian have common noun counterparts as well, e.g. *Fodor* ∼*fodor* ('frill'), *Arany* ∼*arany* ('gold'). Since the word 'fodor' is exceptional, when inflecting it as a common noun, the rule of vowel drop is applied: *fodrot* 'frill.ACC'. However, when inflecting it as a proper name, it is inflected regularly, without dropping the vowel: *Fodort* 'Fodor.ACC'. The common noun 'arany' also has exceptional marking, it is lowering, which means that it has *a* as a link vowel in certain inflectional forms, e.g. in the accusative, instead of the regular bare accusative marker: *aranyat* 'gold.ACC'. But as a proper name, it is inflected regularly: *Aranyt* 'Arany.ACC'. For details on Hungarian morphology see Kornai [1994] and Kenesei et al. [2012]. Psycholinguistic experiments on Hungarian morphology also confirm that proper names are inflected regularly [Lukács, 2001], while common nouns may have exceptional markings.

2.3.3 The Non-compositionality of Proper Names

In order to examine whether proper names are compositional or arbitrary linguistic units, here we give an analysis of how knowledge about the named entity can be deduced from the name[2]. Proper names are not simply arbitrary linguistic units, but they show the arbitrariness most clearly of all, since one can give any name to his/her dog, ship, etc. It follows from the arbitrariness of the initial baptism that proper names say nothing about the properties of the named entity, in fact they do not even indicate what kind of entity we are talking about (a dog, a ship, etc.).

Although *monomorphemic* proper names are classic examples of non-compositionality, they are not semantically empty. For instance, Charlie is a boy by default, but this name is often given to girls in the US, and of course it can be given to pets or products. Semantic implications of proper names (if any) are therefore defeasible. This is in contrast with common nouns, since we cannot call a table 'chair' without violating the Gricean maxims [Grice, 1975]. Monomorphemic proper names have only one non-defeasible semantic implication, namely if one is called *X*, then the predicate 'it is called *X*' will be true (cf. the Millian theory of proper names in Section 2.2).

In the context of the current analysis, two types of *polymorphemic* proper names can be distinguished. First, there are phrases which are headed by a common noun and modified by a proper name, e.g. *Roosevelt square, Columbo pub*. The second type consists of two (or more) proper nouns, e.g. *Theodore Roosevelt, Volvo S70*.

In the case of the former, more frequent type, every non-defeasible semantic implication (except the fact of the naming) comes from the head, the modifier does not make any contribution. This can be shown by removing the head: from the sentence 'You are called from the Roosevelt', one cannot determine the source of the call, which might come from the Roosevelt Hotel, from the Roosevelt College, or from a bar in Roosevelt square. All we have is the trivial implication, that Roosevelt is the name of the place. The fact that the modifier contributes nothing to the semantics of the entire construction can be illustrated better by replacing the proper names with empty elements, e.g. *A square, B pub*. The acceptability of the construction is not compromised even in this case. One further argument against compositionality is that if we try to apply it to polymorphemic proper names, we get unacceptable result: Roosevelt has not lived on Roosevelt square, and Columbo has never been at the Columbo pub.

[2]This subsection is a translated version of a section of the author's article [Simon, 2008].

In the second construction, both head and modifier are proper nouns. The only contribution made by the head to the semantics of the phrase is that we know that the thing referred to by the modifier is a member of the group of things referred to by the head, e.g. Volvo S70 is a kind of Volvo, but not a kind of S70.

Regarding polymorphemic proper names in general, we can say that the head *H* bears the semantics of the entire construction, while the only contribution of the modifier *M* is that it shows that *M* is called 'M' and that it is a kind of *F*. This is in contrast with the classic compositional semantics of common nouns, where the 'red hat' means a hat which is red, the former president used to be a president, etc., and these implications are non-defeasible.

2.3.4 Summary

This section gives an overview how we can distinguish between proper names and common nouns using an approach based in linguistics. The first distinguishing property is the unique reference: common nouns, standing by themselves, never have unique reference. They have to be surrounded by other constituents within a phrase to refer some unique entity in the world, while proper nouns have unique reference on their own. There are, however, proper names which seemingly refer to several entities; it is shown through examples that these do have unique reference. Additional linguistic properties of proper names are presented, based on which a stronger distinction between proper names and common nouns can be made. The distinction based on semantic properties is the clearest: common noun phrases are compositional while proper names are not.

2.4 Conclusion

As can be seen from this overview, the definition of proper names is still an open question in both philosophy and linguistics. If we try to apply the findings presented above to the NER task, we are faced with various challenges. However, there are a few statements which can be used as pillars of defining what to annotate as NEs.

Early works formulate the NER task as recognizing proper names in general. This generality posed a wide range of problems, so the domain of units to be annotated as NEs had to be restricted. In this restricted domain, we find those names (person and place names) which have been postulated as proper names from the very beginnings of linguistics (e.g. in

Plato's dialogue, *Cratylus*, and in Dionysius Thrax' grammar). The third, classical name type (organization names) has been mentioned in grammar books from the 19th century. Although the range of linguistic units to annotate was cut, the challenges have remained, since these kinds of names already exhibit properties which make the NER task difficult.

In the expression 'named entity', the word 'named' aims to restrict the task to only those entities where rigid designators stand for the reference [Nadeau and Sekine, 2007]. Something is a rigid designator if in every possible world it designates the same object and thus has unique reference – unique in every possible world. Rigid designators include proper names as well as species names, mass terms, natural phenomena and measurement units. These natural kind terms are only partially included in the NER task. The MUC guidelines allow for annotating measures (e.g. *16 tons*) and monetary values (e.g. *100 dollars*), which are rigid designators according to Kripke's theory. Some temporal expressions, typically absolute time expressions, are also rigid designators (e.g. *the year 2012* is the 2012th year of the Gregorian calendar), but there are also many non-rigid ones, typically the relative time expressions (e.g. *June* is a month of an undefined year). Thus, the rigid designator theory must be restricted to keep out species names, mass terms and certain natural phenomena, but must also be loosened to allow tagging relative time expressions as NEs.

If we say that every linguistic unit which has unique reference must be annotated as a NE, we should annotate common noun phrases as well. However, dealing with common nouns is not part of the NER task, so other linguistic properties of proper names and common nouns must be considered to make the distinction between them stronger. The greatest difference is the issue of compositionality. Applying Mill's, Saussure's, and Kripke's theory about the meaninglessness of names, we must conclude that proper names are arbitrary linguistic units, whose only semantic implication is the fact of the naming. Thus, the semantics of proper names is in total contrast with the classic compositional semantics of common nouns, as they are indivisible and non-compositional units. To map it to the NER task: embedded NEs are not allowed, and the longest sequences must be annotated as NEs (e.g. in the place name 'Roosevelt square' there is no person name 'Roosevelt' annotated).

There still remain a quite large number of linguistic units which are difficult to categorize. Typically, they are on the border between proper names and common nouns, which is confirmed by the fact that their status varies across languages. We should not forget that the central aim of the NER task is extracting important information from raw text, most of which is contained by NEs. Guidelines should be flexible enough to al-

low the annotation of such important pieces of information. For getting a usable definition of NEs, the classic Aristotelian view on classification, which states that there must be a *differentia specifica* which allows something to be the member of a group, and excludes others, is not applicable. For our purposes, the prototype theory [Rosch, 1973] seems more plausible, where proper names form a continuum ranging from prototypical (person and place names) to non-prototypical categories (product and language names) [Van Langendonck, 2007] (consider the parallelism with the order in which names are mentioned in grammar books). Finally, the goal of the NER application will further restrict the range of linguistic units to be taken into account.

Chapter 3

Handling Metonymic Named Entities

3.1 The Definition of Metonymy

Metonymy is the act of referring to something by the name of something else that is closely related to it. The term is widely used in several disciplines, e.g. in the study of literature it is known as a form of figurative speech (a trope), in rhetorics as a rhetorical strategy, and in linguistics it is often postulated as sense extension. Earlier works on metonymy are focused on distinguishing between metonymy and metaphor (e.g. Fass [1988]). We follow Lakoff and Johnson's cognitive view, namely that the two concepts are quite different: metaphor is "principally a way of conceiving of one thing in terms of another, and its primary function is understanding", while metonymy "has primarily a referential function, that is, it allows us to use one entity to *stand for* another" [Lakoff and Johnson, 1980].

Metonymy is a reference shift: with one linguistic unit we refer not to the primary reference, but to a related one, for example:

(3.1) Just look at all those *hungry mouths* we have to feed.

In Example 3.1, we refer to the whole body (the person) with the name of a part of the body (the mouth). This type of reference shift is systematic, because it can occur with anything which has parts. This kind of metonymy, where a specific part of something is used to refer to the whole, is called synecdoche, or in Lakoff and Johnson's terms, PART FOR WHOLE (henceforth, the name of a certain kind of metonymy is indicated with small capitals, following Lakoff and Johnson's designation). Similar systematic reference shifts can be seen in the following examples:

(3.2) Denise drank the *bottle*. (= the liquid in the bottle → CONTAINER FOR CONTENTS)

(3.3) Ted played *Bach*. (= the music of Bach → ARTIST FOR ARTFORM)

(3.4) Ashe played *McEnroe*. (= tennis with McEnroe → CO-AGENT FOR ACTIVITY)

(3.5) A Mercedes rear-ended *me*. (= my car → CONTROLLER FOR CONTROLLED)

(3.6) The *buses* are on strike. (= bus drivers → OBJECT USED FOR USER)

(3.7) The *book* is moving right along. (= the writing of the book → PRODUCT FOR PROCESS)

These are *conventional* metonymic patterns that operate on semantic classes [Markert and Nissim, 2007a]. In the case of common nouns, such regular shifts have also been called regular polysemy.

There are, however, novel metonymies, created on the fly, which cannot be matched to any pattern:

(3.8) The *ham sandwich* is waiting for his check.

(3.9) Ask *seat 19* whether he wants to swap.

In Example 3.8, the 'ham sandwich' refers to a person, who ordered a ham sandwich in a bar, while in Example 3.9, 'seat 19' refers to the person who is occupying seat 19. These types are called *unconventional* metonymies [Markert and Nissim, 2007a].

3.2 Metonymic Proper Names

In most examples mentioned above, it is common nouns that undergo such reference shifts. Proper names, however, are also likely to occur in metonymies. Most regular metonymic patterns are specific to one particular class (here: person, place, or organization names). Nevertheless, there are some metonymic patterns relevant for all base classes as well. According to Markert and Nissim [2007a], we call them class-specific and class-independent patterns, respectively[1].

[1]Section 3.2 is based mainly on the translation of the author's article [Simon, 2008]. Hungarian examples were adapted to English.

3.2.1 Class-specific Patterns

Metonymy is "using one entity to refer to another that is related to it", as Lakoff and Johnson [1980] define it. Name classes differ in the variety of relations that may serve as the basis for metonymy. In this regard, person names seem to be the most complex. For this reason, unlike previously, we discuss person names only after place and organization names.

Depending on the transparency of the relation between the primary and the contextual reference, common and innovative metonymic patterns can be distinguished. Since the extent of transparency is not the same for everybody, it is hard to be objective in drawing the line between conventional and unconventional metonymies. Therefore, the subsequent list of metonymy categories is not exhaustive: we take into account only the main classes mentioned in the literature [Markert and Nissim, 2007a,b; Lakoff and Johnson, 1980; Simon, 2008].

Place names

Some place names, particularly the names of geo-political entities (countries, states, provinces, etc.), can refer to a location, as well as to its government, its population, or affiliated organizations. In those cases when a place name has an agentive role in the sentence, for example when it makes decisions, has emotions, or causes movement, it is used as a PLACE FOR PEOPLE metonymy, e.g.

(3.10) the *US* position on global warming

(3.11) *Washington* is optimistic

(3.12) the hopeless poverty of *Vietnam*

A place name can occur in many roles, for example, it can stand for the official administration of a country (Example 3.10). A subtype of this kind of metonymy is when the name of the capital stands for the government (Example 3.11). In Example 3.12, the place name refers to the whole or majority of the population.

A widely used metonymy is when a place name is used referring to a sports team affiliated to the place, e.g.

(3.13) two penalty goals by Donaldson preceded *France*'s fifth try

However, this is rather a metonymy chain [Reddy, 1979], because the place name stands for an organization which has members (see ORGANIZATION FOR MEMBERS metonymy below).

Even though PLACE FOR PEOPLE is the most frequent metonymy type in which place names occur, a few other types also exist. One of them is PLACE FOR EVENT, when a location name stands for an event that took place there (Example 3.14). Another one is the PLACE FOR PRODUCT metonymy, in that a place name is used for referring to a product manufactured in the place (Example 3.15).

(3.14) British communists disillusioned with the Soviet Union after *Hungary*

(3.15) a smooth *Bordeaux* that was gutsy enough to cope with our food

Organization names

Similarly to place names, organization names can also refer not only to their primary reference, but also to other references related to it. The most frequent type is ORGANIZATION FOR MEMBERS, especially when a spokesperson acts or speaks on behalf of a group or an organization (Example 3.16), but it also includes cases where all members of the organization participate in an action (Example 3.17).

(3.16) *Renault* sign recycling pact

(3.17) *Microsoft* is writing Windows NT not only for the Intel processor, but for others as well

Organizations typically have a location, so organization names may also stand for the facility that houses them or one of their branches. This is called ORGANIZATION FOR FACILITY metonymy (Example 3.18). An organization may have a value on the stock market, its stock index, to which the name of the organization may also refer: this is the ORGANIZATION FOR INDEX metonymy (Example 3.19).

(3.18) around *Tesco*'s at New Cross

(3.19) *Canon* slips

Another widely used metonymy is ORGANIZATION FOR PRODUCT, where the name of a commercial organization refers to its products (Example 3.20). An organization name can also be used to refer to an event associated with it, this is called ORGANIZATION FOR EVENT (Example 3.21).

(3.20) Tweed may decide to switch that *BMW* for something else

(3.21) in its *Philip Morris* decision in November 1987, the Court held that...

Person names

As can be seen from the examples of metonymies involving place and organization names, metonymies are based on the relation between the primary and the contextual reference. As organizations can have location or stock index, persons can have many properties which stand in some relation to them. However, persons and their properties are more complex than organizations, which is probably the reason why person names usually do not undergo a reference shift *in general*, but in particular roles, e.g. as an artist, or as a co-agent, as it can be seen in Examples 3.3 and 3.4, respectively. Similarly, the CONTROLLER FOR CONTROLLED metonymy can also be used with a person name:

(3.22) *Napoleon* lost at Waterloo.

Radden and Kövecses [1999] call relationships which may give rise to metonymy "metonymy-producing relationships". Such relationships can be: possessing an object, living in a place, being a member of a group, having a particular property, etc.

(3.23) *Peter* is parked on the opposite side of the street

(3.24) the *French* hosted the World Cup Soccer Games

(3.25) every *Tom*, *Dick* and *Harry*

(3.26) he is a *Judas*

Enumerating metonymy-producing relationships for person names could be continued almost indefinitely, but the examples would be far from being conventional metonymies. They rather seem to be examples of innovative language usage.

3.2.2 Class-independent Patterns

Class-independent patterns can be applied to all types of proper names, and even to most nouns. All names can be used as mere signifiers, instead of referring to an object: this metonymic pattern is called OBJECT FOR NAME.

(3.27) *Chevrolet* is feminine because of its sound

This is a classic example of meta-linguistic usage: in Example 3.27, 'Chevrolet' refers to the word 'Chevrolet', as in the last clause we referred to the word used in the example. (According to our assumption, the confusion in terminology around unique reference mentioned in Subsection 2.3.1 is caused by this metonymy being widely used in the literature.)

Another class-independent metonymy type is OBJECT FOR REPRESENTATION, when a proper name refers to a representation (a photo or painting) of the reference of its literal reading. Example 3.28 is metonymic only if somebody is just pointing to a map: in this case 'Malta' refers to the drawing of the island.

(3.28) this is *Malta*

3.2.3 Unconventional Metonymies, Mixed Readings

Unconventional metonymies are non-predictable and context-dependent. No specific category indicating the intended class can be introduced, therefore they do not fit into any of the patterns mentioned above.

(3.29) the bottom end is very *New York/New Jersey* and the top is very melodic

(3.30) funds for Operation Shakespeare had been paid into *Barclays Bank*

In Example 3.29, the location name refers to typical local tunes, while in Example 3.30, the organization name is used as referring to an account at the bank. Both are used rarely, and the comprehension of such expressions highly depends on the context. They are more idiosyncratic than productive patterns, so they can be said to be examples of innovative, novel language use (cf. Subsection 3.2.1).

In addition to literal and metonymic readings, there are examples where two predicates are involved, each inducing a different reading, resulting in *mixed reading*. This occurs often with coordinations and appositions.

(3.31) countries paying money to users of contraception: *Bangladesh, Egypt*, ...

(3.32) *BT*, Britain's main telephone company, announced a 36% fall

In Example 3.31, country names have mixed reading: on the one hand, it is a PLACE FOR PEOPLE metonymy, where the word 'countries' refers to the official administration of the countries, and Bangladesh and Egypt are some of these countries. On the other hand, due to the construction where 'Bangladesh' and 'Egypt' are items in the list of countries, they have a simple literal reading. In Example 3.32, 'BT' refers to a spokesperson who made an announcement, thus it is an ORGANIZATION FOR MEMBERS metonymy. However, 'BT' and 'Britain's main telephone company' are in apposition, the latter defining the former. Because of this kind of definition, 'BT' may even have a literal reading.

3.3 Metonymy Resolution in NLP

As our overview of metonymic proper names shows, NEs are ambiguous referential elements of discourse. Metonymic usage of NEs is frequent in natural language, therefore the resolution of metonymic NEs would be a direct benefit to NER and indirectly to all NLP tasks that require NER. The importance of resolving metonymies has been shown for a variety of NLP tasks, such as MT [Kamei and Wakao, 1992], question answering [Stallard, 1993], anaphora resolution [Markert and Hahn, 2002], and IR [Leveling and Hartrumpf, 2006].

However, metonymy is known to be difficult for NLP for the simple reason that conceptual mappings between related references are not linked to particular linguistic forms. Thus, classic compositional semantic analyses using a static lexicon are inadequate in the case of metonymic NEs, since the latter often deal with word senses that are not listed in the lexicon.

Early works, lacking corpora annotated for metonymy, are based on largely manually constructed example lists, and are therefore often biased to make a particular point of interest (e.g. Lakoff and Johnson [1980]; Pustejovsky [1995]). First attempts on computational resolution of metonymies are mostly based on inference rules (e.g. Fass [1988]). Work in this vein takes the view that figurative language processing should not be approached as a language related phenomenon, but as a problem for a general reasoning ability. The lack of language resources is the main cause that early computational works are evaluated in comparison to constructed examples only (e.g. Fass [1988]; Hobbs et al. [1993]), or, though using naturally-occurring data, based on subjective intuition (e.g. Harabagiu [1998]; Stallard [1993]). Results are thus hardly comparable as they all operate within different frameworks.

Markert and Nissim [2002] postulate metonymy resolution as a class-based WSD task for the semantic class of locations, and later for the class of organizations as well [Markert and Nissim, 2007a]. Their works were later extended to form a SemEval-2007 shared task [Markert and Nissim, 2007b], for which metonymically annotated datasets were provided. Many NLP tasks have benefitted enormously from shared task evaluations, competitions which have significantly improved the state-of-the-art. The situation is similar in the case of the metonymy resolution task: the existence of a reference dataset gave great impetus to the creation of a wide variety of metonymy resolution systems.

Since the possible interpretations of a potentially metonymic word (PMW) can be viewed as corresponding to the word's possible senses, metonymy resolution can be interpreted as a classification task, for which supervised machine learning techniques can be used. Participants of the shared task and later users of the SemEval dataset use a range of supervised learning paradigms, e.g. maximum entropy [Farkas et al., 2007], decision trees [Nicolae et al., 2007], and Support Vector Machines (SVMs) [Ferraro, 2011]. Most systems use shallow features extracted directly from the training data (parts-of-speech, co-occurrences, and collocations), and morphological and syntactic features. Almost all systems use external resources: lexical databases, such as WordNet [Farkas et al., 2007; Nastase and Strube, 2009], other corpora [Brun et al., 2007] or the Web as corpus [Farkas et al., 2007], or encyclopedic knowledge extracted from Wikipedia [Nastase and Strube, 2009; Judea et al., 2012].

3.3.1 SemEval-2007 Metonymy Resolution Task Description

Here we give a short overview of the metonymy resolution shared task of SemEval-2007. We only provide information that is needed for the interpretation of our results shown in Subsection 3.3.2. For more details see the task description paper by Markert and Nissim [2007b].

The dataset consists of samples extracted from the British National Corpus (BNC), Version 1.0. Samples contain four sentences: the sentence in which the PMW occurs, two before, and one after.

The shared task focused on two NE classes, `location` and `organization`, each corresponding to a subtask. For both subtasks, random subsets of samples were selected as training and test datasets.

Metonymy annotation was performed by using categories of conventional metonymies described in Subsection 3.2.1. Class-independent

metonymic readings (see Subsection 3.2.2) were applied for both location and organization names. In addition, instances of mixed readings and unconventional metonymies (see Subsection 3.2.3) were also annotated (the latter as `othermet`). The reading distribution of training and test sets for both subtasks are shown in Tables 3.1 and 3.2. Percentages are provided by Ferraro [2011]. Readings are sorted according to their frequency.

reading	train	test
literal	737 (79.68%)	721 (79.41%)
place-for-people	161 (17.41%)	141 (15.53%)
mixed	15 (1.62%)	20 (2.20%)
othermet	9 (0.97%)	11 (1.21%)
place-for-event	3 (0.32%)	10 (1.10%)
object-for-name	0 (0%)	4 (0.44%)
place-for-product	0 (0%)	1 (0.11%)
object-for-representation	0 (0%)	0 (0%)
total	925	908

Table 3.1: Reading distribution for locations in the SemEval-2007 datasets.

In addition to metonymy annotation, several types of linguistic annotation were also provided by the organizers for both training and test sets. This included the BNC tokenization, part-of-speech (POS) tags, and manually annotated dependency relations for each PMW.

The location and organization subtasks were further divided into three subtasks of different granularity levels, resulting in six subtasks for which participants were allowed to submit their results. The fine-grained evaluation aimed at distinguishing between all categories, while the medium-grained evaluation grouped different types of metonymic usage together and addressed literal/mixed/metonymic usage. The coarse-grained subtask was in fact a literal/non-literal two-class classification task.

For each target category, precision, recall and F-measure ($\beta = 1$) were counted. (For more details on the standard evaluation methods, see Subsection 5.3.1.) As a baseline method, assignment of the most frequent category label (`literal`) was used for each subtask.

reading	train	test
literal	690 (63.30%)	520 (61.76%)
org-for-members	220 (20.18%)	161 (19.12%)
org-for-product	74 (6.79%)	67 (7.96%)
mixed	59 (5.41%)	60 (7.12%)
org-for-facility	15 (1.38%)	16 (1.90%)
othermet	14 (1.28%)	8 (0.96%)
object-for-name	8 (0.73%)	6 (0.71%)
org-for-index	7 (0.64%)	3 (0.36%)
org-for-event	2 (0.18%)	1 (0.12%)
object-for-representation	1 (0.09%)	0 (0%)
total	1090	842

Table 3.2: Reading distribution for organizations in the SemEval-2007 datasets.

3.3.2 GYDER: System Description

We built a maximum entropy metonymy resolution system, named GYDER (the acronym was formed from the initials of the authors' first names), which was submitted to the SemEval-2007 metonymy resolution shared task, and achieved the highest scores. In the subsequent description we discuss feature engineering and present our results[2].

GYDER uses the same maximum entropy toolkit as our general-purpose NER system does[3], setting Gaussian prior to 1. Due to the small number of instances and features, the learning algorithm always converged before 30 iterations, so the evaluation process only took seconds. (For more details on our general-purpose NER system, see Chapter 5. That chapter discusses maximum entropy learning, as well as Gaussian prior, iteration, and further issues concerning supervised machine learning.)

We also tested the classic C4.5 decision tree learning algorithm [Quinlan, 1993], but our early experiments showed that the maximum entropy learner was consistently superior to the decision tree classifier for this task, yielding about 2-5% higher accuracy scores on average in all of the subtasks (on the training set, using cross-validation).

[2]This subsection is based mainly on our article published in SemEval proceedings [Farkas et al., 2007].

[3]http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Feature engineering

We tested several features describing orthographic, syntactic, or semantic characteristics of PMWs. We mainly followed Nissim and Markert [2005], who reported three classes of features as being the most relevant for metonymy resolution: the annotation of dependency relations, the type of the PMW's determiner, and the plurality of the PMW. We also report on some features that did not work.

We used *dependency information* in several ways. The simplest one was using the type of dependency relation and the word form of the related word as features assigned to the PMW (in the case there were more related words, each of them became a feature). As determined by evaluation on the test set, adding the type of dependency relation as a feature caused significant improvement in the organization subtask and slight improvement in the location subtask.

To overcome data sparseness, using features which generalize from individual words seemed useful as well. Three different methods were used to achieve this, each incorporating the information of dependency relations.

After analysing metonymic occurrences of NEs in the training dataset, we conclude that metonymic usage is usually induced by the predicate on which the PMW depends. First, we manually collected a list of *trigger words* that are likely to appear close to the PMW. Each indicator word was assigned to one of several semantic subclasses, which were created based on the annotation guidelines provided by the organizers. For example, verbs like 'announce', 'say', 'declare' are members of the semantic subclass of communication actions, which is typical of the ORGANIZATION FOR MEMBERS metonymy. If the PMW's predicate appeared in the list, its semantic subclass was assigned to the PMW as a feature. This is the only resource in our final system that was manually built. When measuring performance on the test set, omitting this feature did not change the accuracy in the organization subtask and decreased accuracy by 0.44% in the location subtask.

Second, we used *Levin's verb classes* [Levin, 1993] to generalize words of the most relevant dependency relations (subject and object). The added feature was the number representing the class automatically extracted from Levin's verb classification index[4]. Unfortunately, this feature only caused insignificant improvement.

Third, we gathered the hypernym path from *WordNet* [Fellbaum, 1998]

[4] `http://www-personal.umich.edu/~jlawler/levin.html`

for each related word's sense #1. Based on these paths, synsets whose tree contained a node that frequently indicated metonymic sense were collected. If the word appeared in one of the collected subtrees, it received this kind of feature.

Following Nissim and Markert [2005], we distinguished between definite, indefinite, demonstrative, possessive, wh and other *determiners*. We also marked cases where the PMW was sentence-initial, and thus necessarily determinerless. However, without complete syntactic analysis, assigning determiners to PMWs was not straightforward. After some experiments, we decided to link the closest determiner and the PMW together in those cases when only adjectives, if anything, were found between them in the sentence. This feature was only useful for the resolution of organization PMWs.

The *plurality* feature was particularly useful to find instances of the ORGANIZATION FOR PRODUCT metonymy. To decide whether a PMW is in plural form, we used the web as corpus. For each PMW ending with the letter 's', two Google queries were run: one for the complete word form, and one for the word without its last character. If the number of hits for the shorter form was significantly higher than one for the complete form, the plurality feature was assigned to the PMW[5].

Additionally, we included the *surface form* of the PMW as a feature. Cross-validation on the training corpus showed that the use of this feature causes a 1.5% improvement in F-measure for organizations and a slight decrease for locations, so it was used only for the organization subtask. Our decision was confirmed by the results on the test set as well.

During development, we used random five-fold cross-validation on the training dataset to determine the usefulness of each feature (see Subsection 5.3.1 for details on cross-validation). A few of them was proved to be *unsuccessful features*, and were not included in the submitted system. Their list is as follows: automatically collected words which frequently appear close to metonymic PMWs in the training dataset, NE labels assigned by a state-of-the-art NER system [Szarvas et al., 2006b], POS tags around PMWs, capitalization and other surface characteristics for the PMW and nearby words, and the inflectional category of the verb closest to the PMW in the sentence.

[5]This feature and similar web-as-corpus approaches for lemmatization of NEs are detailed in one of the co-authors' PhD thesis [Szarvas, 2008]

Results

The assigment of the most frequent category label (`literal`) to every PMW in the training dataset can serve as a baseline when evaluating the system (baseline1 in Table 3.3, cf. training dataset percentages in Tables 3.1 and 3.2). As another baseline, we evaluated the system without the WordNet, Levin, trigger, and PMW word form features (baseline2 in Table 3.3). This baseline system is quite similar to that described by Nissim and Markert [2005], so we get the discriminative power of our four new features. Table 3.3 shows the results (F-measure) of the baseline systems and our submitted system on the fine granularity level.

run	baseline1	baseline2	submitted
org train 5-fold	63.30	77.51	80.92
org test	61.76	70.55	72.80
loc train 5-fold	79.68	85.58	88.36
loc test	79.41	83.59	84.36

Table 3.3: Results of the baseline systems and our submitted system on the fine granularity level.

For the medium and coarse subtasks of the location domain, we simply generalized the fine-grained results, i.e. the counts of classes `place-for-x` and `othermet` (and `mixed` on the coarse level) were summed. In the organization domain, we merged the fine-grained classes into one metonymic class before training. Overall F-measure for each domain/granularity can be seen in Table 3.4. In general, coarse-grained evaluation did not show a significantly higher performance, proving that the main difficulty is distinguishing between literal and metonymic usage, not separating metonymy types from each other. Thus, data sparseness remained a problem for coarse-grained classification as well.

	coarse	medium	fine
location	85.24	84.80	84.36
organization	76.72	73.28	72.80

Table 3.4: Overall F-measure of the GYDER system for each domain/granularity.

Per-class results (precision, recall, F-measure) of the submitted system for location and organization domains are shown in Tables 3.5 and 3.6. In the location subtask, our system never predicted values for the four small classes `place-for-event`, `place-for-product`, `object-for-name`, and `other`, as these had only 12 instances altogether in the train dataset. Since our system's performance on the `mixed` category also remained low, we decided to simplify the location subtask as a binary classification task between `literal` and `place-for-people` categories. Results are similar in the organization subtask: while the system ignored three of the smallest categories `othermet`, `organization-for-index`, and `organization-for-event` (a total of 12 instances), we achieved meaningful results for the six major categories. (The class `object-for-representation` remained undefined in both location and organization subtasks, since it has only one instance altogether in the train and test set, and none of the submitted systems could produce results for it.)

reading	#	prec	rec	f
literal	721	86.83	95.98	91.17
place-for-people	141	68.22	51.77	58.87
mixed	20	25.00	5.00	8.33
othermet	11	-	0.0	-
place-for-event	10	-	0.0	-
object-for-name	4	-	0.0	-
place-for-product	1	-	0.0	-

Table 3.5: Per-class results of the GYDER system for location domain.

Conclusions

Tables 3.5 and 3.6 show results for all classes sorted by F-measure. Comparing them with Tables 3.1 and 3.2, which show the reading distribution for each class sorted by frequency, it can be seen that the order is very similar. Thus, several categories do not contain a sufficient number of examples for machine learning. For this reason, we decided early to accept the fact that these categories will not be learned and to concentrate on those classes where learning seemed feasible. After simplifying the task in such a way, data sparseness still remained a problem. It can be eased by

reading	#	prec	rec	f
literal	520	75.76	90.77	82.59
org-for-members	161	65.99	60.25	62.99
org-for-product	67	82.76	35.82	50.00
mixed	60	43.59	28.33	34.34
object-for-name	6	50.00	16.67	25.00
org-for-facility	16	100.0	12.50	22.22
othermet	8	-	0.0	-
org-for-index	3	-	0.0	-
org-for-event	1	-	0.0	-

Table 3.6: Per-class results of the GYDER system for organization domain.

application of several generalization approaches, like grouping words implicating metonymic usage into semantic subclasses, as we did using the Levin and trigger features. Poibeau [2007] follows a similar way of reducing search space size, but with a different approach, collecting patterns for separating literal usage of NEs from metonymic readings.

The maximum entropy method and the features we selected performed well enough to achieve the best scores in all six subtasks of the shared task, as can be seen in Table 3.7 reporting F-measure for all participating systems. Our four new features (WordNet, Levin, trigger and the PMW's word form) proved to be discriminative features for metonymy resolution, as indicated by results in Table 3.3, and we believe they are useful in general.

	baseline	FUH	UTD-HLT-CG	XRCE-M	GYDER	up13
loc-coarse	79.4	77.8	84.1	85.1	85.2	75.4
loc-medium	79.4	77.2	84.0	84.8	84.8	75.0
loc-fine	79.4	75.9	82.2	84.1	84.4	74.1
org-coarse	61.8	-	73.9	73.2	76.7	-
org-medium	61.8	-	71.1	71.1	73.3	-
org-fine	61.8	-	71.1	70.0	72.8	-

Table 3.7: Results of all participating systems for all subtasks.

To prove the usefulness of our features, we collected some results which have been published since the SemEval-2007 shared task.

Nastase and Strube [2009] use three types of features. First, they take the minimum set of necessary features presented by Nissim and Markert [2005]. Second, following the generalization approach taken by us and other participating teams, they use WordNet and the page and category network of Wikipedia to assign supersenses to PMWs. Third, our observation that using the word form of PMWs as features leads to improvement in determining the reading for organization names, and the observation of Brun et al. [2007] that certain locations are more likely to be used with an event reading than other locations, lead them to mine for certain pieces of information in Wikipedia relations, and to add them as features for PMWs.

Ferraro [2011] uses syntactically and semantically based features as well. The former is very similar to Nissim and Markert's features [Nissim and Markert, 2005], while the latter consist of WordNet relations and a word-scoring function aimed at extracting the underlying conceptual meaning of PMWs. He tested several learning algorithms, concluding that SVMs provide the most predictive power, so it was used to evaluate his system on the SemEval-2007 test dataset.

Judea et al. [2012] provide a method for the derivation of distributional semantic representations based on Wikipedia and WikiNet. The resource obtained was evaluated through metonymy resolution.

task ↓ / system →	Nastase-Strube	Judea et al.	Ferraro	GYDER
loc-coarse	**86.1**	-	83.4	85.2
loc-medium	**85.9**	85.6	82.5	84.8
loc-fine	**85.0**	-	82.0	84.4
org-coarse	74.9	-	75.5	**76.7**
org-medium	72.4	72.0	69.9	**73.3**
org-fine	71.0	-	69.0	**72.8**

Table 3.8: Results of systems which have been published since the SemEval-2007 shared task, compared to GYDER's scores.

Table 3.8 summarizes the results of the systems described above. (Judea et al. [2012] did not provide results on the coarse and fine granularity level.) The highest F-measure for each subtask is in bold type. In the location subtask, the systems of Nastase and Strube [2009] and Judea et al. [2012] outperform GYDER, but in the organization subtask our scores are still the highest.

3.4 Conclusion

Metonymy is using one word to refer to another that is related to it. As can be seen from our overview, some NEs are also likely to occur in metonymies. Although the relation between the primary and the contextual reference is not completely transparent, we made a distinction between conventional and unconventional metonymies, based mainly on Markert and Nissim [2007a]. In the case of the former, reference shift is applied for a semantic class, in our case for location and organization names. Metonymies of the latter type are rather examples of novel, innovative language use, which are hard to recognize for both human annotators and NLP systems.

Based on the results of our supervised metonymy resolution system, we can conclude that the main borderline does not lie between conventional and unconventional metonymies, but rather between literal and metonymic usage. Our system did not emit any labels for the `othermet` category, either for locations or organizations. Moreover, it did not emit labels for classes with a small number of instances, such as `place-for-event` or `organization-for-index`. The identification of metonymies is based more on their frequency than on any other properties.

Our system provided the best results for metonymic classes which are journalistic cliches, such as `organization-for-members` and `place-for-people`, incorporating PLACE FOR SPORTS TEAM and CAPITAL FOR GOVERNMENT metonymies. Instances of these classes cover most of metonymic NEs, particularly in newswire. We therefore believe that the metonymy resolution task should be simplified as a task of recognizing literal readings and journalistic cliches. This type of classification would directly benefit NER and indirectly all NLP tasks that require NER.

Supervised systems have two main disadvantages: they do not allow the recognition of new classes besides those pre-defined, and they require a large amount of texts annotated with specific linguistic information. Thus, approaching the metonymy resolution task with supervised systems for recognizing new metonymy types does not work. Since conceptual mappings between related references of metonymic words are not linked to particular linguistic forms, corpora with rich semantic annotation is needed for the task. Although using some surface and syntactic information leads to improvement in resolving metonymies, future systems should exploit more semantic knowledge, or the power of a larger dataset, or preferably both.

Chapter 4

Gold and Silver Standard Corpora for Named Entity Recognition

The statistical approach to NLP requires large amounts of text, i.e. corpora. A *corpus* is "a collection of pieces of language text in electronic form, selected according to external criteria to represent, as far as possible, a language or language variety as a source of data for linguistic research" [Sinclair, 2005]. Machine learning algorithms typically learn their parameters from corpora, and systems are evaluated by comparing their output to another part of the corpus, or to another corpus. Corpora can be classified according to various criteria, including:

a. source: written, spoken, or originally in electronic format;

b. domain: newswire, academic, literature, etc.;

c. language: monolingual (e.g. English), bilingual (e.g. English–Hungarian), or multilingual;

d. location: e.g. the English of UK, the US or Canada;

e. date: e.g. codices from the Old Hungarian period, or present-day standard Hungarian.

To be a *gold standard corpus*, a dataset must meet ideally all, but at least most of the following requirements. In the case of dead languages or highly specialized sublanguages, it must be exhaustive. In the case of living languages, a corpus cannot be exhaustive, but it must aim for representativeness, which can be ensured by balanced sampling of sources and/or domains of the text. It must be split into language units, for example sentences and tokens, depending on the task it is intended to be used

for. Some linguistic information must be assigned to each language unit as annotation. The annotation must be done or at least corrected by human annotators following the rules of some annotation guidelines. Gold standard corpora usually have pre-defined size, since continuous manual annotation is not feasible.

A *silver standard corpus*, or at least the annotation in it is automatically generated, without any human intervention. For this reason, it is extendable with new texts and/or annotation levels, and is not restricted in size. Automatically generated corpora can be quite useful for improving performance of NLP systems in several ways.

In this chapter, we first give an overview of corpus building in general, see Section 4.1. In Section 4.2 we introduce the gold standard datasets used regularly in the field of NER. Since gold standard corpora used in NER are in most cases highly domain specific, containing mostly newswire text, and are restricted in size, in Section 4.3 we present a way of automatically generating NE tagged corpora as an efficient alternative of gold standard corpora.

4.1 Corpus Building

In this section, we give a brief overview of the main issues concerning corpus building. There are excellent textbooks on corpus linguistics, e.g. McEnery and Wilson [2001]; Lüdeling and Kytö [2008]; O'Keeffe and McCarthy [2010], to mention only a few, so we direct the reader to them for more complete discussions.

The very first step of corpus building is determining *corpus design*: intended uses of the corpus, the language variety to be covered, the domain(s) to be represented, the required size, and the future access of the corpus. The last criterion is of key importance in text collection, if one intends to build a corpus that is freely accessible, at least for research purposes. In addition to the effort put in data processing, a considerable amount of time has to be devoted to acquiring texts and clearing copyrights. The data collection process and negotiations on Intellectual Property Rights (IPR) matters may drag on for months. In corpus linguistics textbooks, issues of data collection and more specifically copyright clearance are hardly touched upon. However, there are a few current attempts at improving the state-of-affairs, e.g. Clercq and Perez [2010]; Xiao [2010].

A corpus is a well-organized collection of data, "collected within the boundaries of a *sampling frame* designed to allow the exploration of certain linguistic feature (or set of features) via the data collected" [McEnery,

2004]. If the object of study is a highly restricted sublanguage or a dead language, identifying the texts to be included in the corpus is straightforward. For example, when constructing the Old Hungarian corpus [Simon et al., 2011; Simon and Sass, 2012], we had to acquire all available sources from the Old Hungarian period (896–1526), creating a corpus of fixed size (approx. 2 million tokens). However, moving forward in the history of the Hungarian language, after the time of Gutenberg's invention of the printing press, the amount of textual sources increases to the point where including all of them in a corpus seems impossible. In such cases, a sampling frame is of crucial importance. The corpus should aim for balance and representativeness within a specific sampling frame, in order to allow a particular variety of language to be studied.

The issue of *representativeness* is one of the most frequently discussed questions of corpus design (e.g. Biber [1993]). However, attempting to reach representativeness is like shooting at a moving target. Quoting Hunston [2008]: "representativeness is the relationship between the corpus and the body of language it is used to represent". But what do we know about the body of language? Getting information about the language is the very reason why we build corpora. Maybe it is easier to give examples of unrepresentativeness. If one wants to build a representative general-aim corpus of present-day standard Hungarian, collecting only blogs about sports will not be enough. Or citing McEnery's example [McEnery, 2004]: imagine that a researcher decides to construct a corpus to assist in the task of developing a dialogue manager for a telephone ticket selling system. This researcher will not sample the novels of Jane Austen or movie subtitles to cover the language usage of phone dialogues. Thus, representativeness is a goal we can aim for, without being convinced that we will reach it.

The first phase of corpus building work starts with the *acquisition of source data*. In the case of written texts, there are three methods to achieve this. In the most fortunate case, the corpus consists of texts which are available electronically, in some machine-readable format. If a source is only available in print, digitization is necessary in the form of mainly manual scanning followed by a conversion process from the scanned images into regular text files aided by optical character recognizer software. This step involves extensive manual proofreading and correction to ensure initial resources of good quality as input to further computational processing. The third method is typing up text by hand, which is usually avoided unless the texts concerned are not available in any other way. This is the case, for example, with old manuscripts, handwritten letters, or codices [Oravecz et al., 2010].

McEnery and Wilson [2001] describe annotated corpora as being "enhanced with various types of linguistic information". The *development of annotation*, which is more or less prototypical in modern language corpora, requires a number of standard NLP tasks: sentence segmentation and tokenization, morphological analysis and morphosyntactic disambiguation. These basic processing steps are usually carried out automatically, since tokenizers, sentence splitters and POS taggers are reliable enough for certain languages such as English and Hungarian that a wholly automated annotation is feasible. Error rates associated with taggers are low, typically reported at around 3 percent. For example, the automatic morphosyntactic annotation in the Hungarian National Corpus reaches a general precision of about 97.5%, i.e. 2.5% of all wordforms has an erroneous analysis [Váradi, 2002]. Higher precision could only be achieved by manual annotation, which is usually not feasible for large amounts of data.

More typically, however, NLP tools are not sufficiently accurate so as to allow for fully automated annotation. In these cases, semi-manual or fully *manual annotation* is required. For building a highly accurately annotated corpus, first the annotation scheme has to be developed, then annotation guidelines have to be taught to the annotators. The more elaborated the guidelines, the cleaner and more useful the corpus as long as the guidelines remain teachable, but when they become too complex, annotators begin to perform at an unacceptably high error rate. Guidelines have to define the annotation task, enumerate the types of language units to annotate, and give examples of what to annotate and what not to annotate. (For examples of annotation guidelines for the NER task, see Section 2.1.)

As we proceed in enriching the corpus with linguistic annotation from the basic processing steps to more difficult semantic annotation levels, it will be clear that the more linguistic and semantic knowledge is required, the more liquid the annotation process gets. There are some linguistic phenomena which are hard to define, as can be seen in the case of NEs (see Chapter 2) and metonymies (see Chapter 3). If the guidelines are not accurate enough, such linguistic phenomena will be identified and categorized based on intuitions of the annotators. This strategy may be unproblematic for very clear-cut classes, but an exhaustive annotation will confront the researcher with many cases that are not clear-cut. In such cases, *inter-annotator agreement* is usually measured. The most simple measure is the joint probability of agreement:

$$\frac{2 * |\text{identically tagged entities}|}{|\text{entities tagged by annotator A}| + |\text{entities tagged by annotator B}|}$$

This formula was used for calculating inter-annotator agreement in the case of finding and labelling metaphorical expressions in a corpus we built for a study of literal versus metaphorical language use [Babarczy et al., 2010b]. At the first attempt, inter-annotator agreement was only 17%. After refining the annotation instructions, we made a second attempt, which resulted in an agreement level of 48%, which is still a strikingly low value. These results indicate that the definition of metaphoricity is problematic in itself, and that the refinement of annotation guidelines results in a more accurate annotated dataset. Contrary to finding metaphorical expressions, recognizing NEs in texts usually results in much higher inter-annotator agreement, mostly above 90%.

However, the joint probability of agreement does not take into account that agreement may happen solely based on chance. For this reason, other coefficients such as Cohen's κ and Krippendorff's α are traditionally more often used in CL [Artstein and Poesio, 2008]. The strength of agreement is said to be perfect above 0.8 κ value, according to Landis and Koch [1977].

When building a gold standard corpus, researchers aim for as high inter-annotator agreement as possible, so samples whose annotation cannot be agreed on are often excluded from the corpus (e.g. Markert and Nissim [2007b]), resulting in clean and noiseless corpora. NLP has been predominantly focused on relatively small and well-curated datasets; there are, however, new emerging attempts at processing data in non-standard sublanguages such as the language of tweets, blogs, social media, or historical data. In these NLP tasks, models based on cleaned data do not perform well, so researchers started using collaboratively constructed resources to substitute for or supplement conventional resources such as linguistically annotated corpora.

4.2 Gold Standard Corpora for Named Entity Recognition

This section presents gold standard corpora used in the field of NER, examining aspects such as types of entities covered, domains preferred, languages supported, and size. We do not claim this overview to be exhaustive, rather, we focus on corpora available for English (since these have received the greatest attention in the NER research) and Hungarian (since our research activity was mostly concerned with them). For English, we discuss freely available datasets which were built for shared tasks and became the major standards in NER: MUC-7 [Chinchor, 1998b] and CoNLL-

2003 [Tjong Kim Sang and De Meulder, 2003] datasets. As for Hungarian, we examine the Szeged NER corpus [Szarvas et al., 2006a] and the Criminal NE corpus[1], and finally say a few words about the HunNer corpus [Simon et al., 2006].

4.2.1 The Entity Type Factor

One of the main properties of corpora is the annotation scheme they follow, which determines the range of NE types annotated in them. Since Section 2.1 gives an overview of annotation schemes applied in the NER task, here we only mention some approaches and show the difference between the major annotation schemes.

As discussed in Chapter 3, some NEs may have metonymic readings in certain contexts, which raises certain questions even at the level of annotation. There are two approaches to follow. First, one can always tag a NE according to its contextual reference. In this case, the 'White House' in Example 4.1 would be tagged as an organization name. This rule is called *Tag for Meaning*, and is applied e.g. by LDC in the LCTL project [Linguistic Data Consortium LCTL Team, 2006].

(4.1) the *White House* announced...

The second approach is called *Tag for Tagging*, when NEs are always tagged according to their primary reference, regardless of the context. Following this rule, the 'White House' in Example 4.1 would be tagged as a location name. Most of the major annotation schemes use this rule. A combination of the two approaches is probably the best solution, namely annotating metonymic cases with tags which provide information about the primary reference as well as the contextual reference (here, `LOC:ORG`). The creators of the Criminal NE corpus built two annotated versions of the corpus: one following the Tag for Meaning rule, and the other one according to the Tag for Tagging approach. This solution offers the possibility of handling metonymicity at higher processing levels, e.g. in anaphora resolution, while provides interoperability between various annotation schemes.

As shown in Section 2.1, several different annotation schemes exist in the field of NER. The MUC-6 standard uses tags for person, organization, and location names, date and time expressions, monetary values and percentages. In addition to them, MUC-7 introduces the `Measure` tag. The CoNLL NER shared tasks of 2002 and 2003 focused on marking tags for

[1] `http://www.inf.u-szeged.hu/rgai/nlp?lang=en&page=corpus_ne`

the three basic types (PER, LOC, ORG), and MISC. LDC, in its LCTL project, also tags titles beyond the basic categories. More recent works, aiming at higher level processing tasks, expanded it into fine-grained categorical hierarchies. BBN categories [Brunstein, 2002] are used for question answering and consist of 29 types and 64 subtypes. Sekine's extended hierarchy [Sekine et al., 2002] is made up of 200 subtypes, while in the ACE annotation scheme [ACE, 2008] seven types and 43 subtypes are distinguished.

Researchers attempting to merge these datasets to get a bigger training corpus are faced with the problem of combining different tagsets and annotation schemes. Automatically generated corpora also require gold standard datasets to be evaluated against. In this case, researchers also have to resolve incompatibility issues. Different tagsets can be merged only if some tags are removed or mapped to common types.

4.2.2 The Domain Factor

Early works in IE and NER focused on extracting events and concerned NEs from military reports. Attention then turned to the processing of journalistic articles. However, the topic of news reports used as training and test dataset in the first MUCs still remained similar, including terrorist activities, airplane crashes and rocket launches. Organizers of more current shared tasks moved from military to civil topics: the CoNLL datasets consist of newspaper articles, and the ACE evaluation also included several types of informal text styles such as weblogs and text transcripts from telephone speech conversations.

Although several topics were investigated since then, such as technical emails [Poibeau and Kosseim, 2001], religious texts and scientific books [Maynard et al., 2001], the datasets created for these topics are not freely available, so they cannot serve as reference corpora. Thus, freely available NE tagged corpora remain highly domain-specific.

The multilingual NER evaluation in MUC-6 was run using training and test articles from comparable domains for all languages. However, in MUC-7, organizers changed the domains between the development and test sets, which caused similar effects across languages. Participants expressed disappointment upon comparing test scores to development scores [Chinchor, 1998b]. In recent years, investigating the impact of domain became one of the major research topics in NER. Experiments [Poibeau and Kosseim, 2001; Maynard et al., 2001; Ciaramita and Altun, 2005] demonstrated that although any domain can be reasonably supported, porting a system to new domains remained a major challenge.

Nothman et al. [2008] evaluated MUC-7 and CoNLL-2003 datasets, and BBN Pronoun Coreference and Entity Type Corpus [Weischedel and Brunstein, 2005] against each other. They used the C&C Maximum Entropy NE tagger [Curran and Clark, 2003] with default orthographic, contextual, in-document, and first name gazetteer features. After merging the various tagsets, training and testing was run both with and without the `Miscellaneous` category.

	with `MISC`		no `MISC`		
	CoNLL	BBN	MUC	CoNLL	BBN
MUC	-	-	**74.4**	51.7	54.8
CoNLL	**81.2**	62.3	58.8	**82.1**	62.4
BBN	54.7	**86.7**	75.7	53.9	**88.4**

Table 4.1: Cross-domain test results for MUC-7, CoNLL-2003 and BBN corpora.

As shown in Table 4.1, each set of gold standard training data leads to significantly higher performance on corresponding test sets (with boldface) than on test sets from other sources. The ca. 20-30% decrease in overall F-measure confirms that the training corpus is an important performance factor. Cross-domain evaluation usually gives low performance results, as can be seen even from our results of training on a silver standard corpus automatically generated from Wikipedia, and then testing the model against newswire gold standard corpora (see Subsection 4.3.4).

4.2.3 The Language Factor

As discussed earlier, IE and NER have been in the focus of numerous open competitions in the USA since the 1990s, primarily organized by the government-sponsored organizations Defense Advanced Research Projects Agency (DARPA) and NIST. These competitions have significantly improved the state of the art, but their focus has mostly been on the English language. However, a good proportion of work in NER research addresses the questions of language independence and multilingualism.

Certain languages are particularly interesting from the point of view of NER. For example, in German, not only proper names are capitalized, but every noun, so the capitalization feature does not have as much discriminative power as in English. Besides English, German was also a target

language of CoNLL-2003, where significantly lower overall F-measures were reported for the latter [Tjong Kim Sang and De Meulder, 2003]. Some other feature types become useless in the case of CJKV languages because of the different writing systems. However, they are well studied, and NER systems for these languages reach similar scores as the state-of-the-art systems for English [Merchant et al., 1996]. Most recently, Arabic (e.g. Benajiba et al. [2008]) has started to receive a lot of attention, mainly for political reasons. For NER systems for other languages see the survey of Nadeau and Sekine [2007].

As a highly agglutinative language, *Hungarian* NER also poses its own challenges. Because of its rich morphology, features based on morphological information are quite important. Therefore, a gold standard corpus for Hungarian NER should contain rich morphosyntactic information.

Mapping standard tagsets and adapting annotation schemes used for English NER to Hungarian also raises a few issues. There are language units which are considered NEs by the CoNLL annotation scheme, but are not considered NEs in Hungarian. These are typically the ones being on the border between proper names and common nouns, and their usage varies from language to language, i.e. the non-prototypical categories (cf. Section 2.4): names of languages, nationalities, religions, political ideologies; adjectives derived from NEs; names of months, days, holidays; names of special events and wars.

The first gold standard NE tagged corpus for Hungarian was the *Szeged NER corpus* [Szarvas et al., 2006a] created by researchers at the University of Szeged. It is a subcorpus of the Szeged Treebank [Csendes et al., 2004], which contains full syntactic annotation created manually by linguist experts. A significant part of these texts has been annotated with NE class labels in line with the annotation scheme of the CoNLL-2003 shared task. The corpus consists of short business news articles collected from Magyar Távirati Iroda, the Hungarian news agency.

Since the Szeged NER corpus is highly domain-specific, the need emerged for a large, heterogeneous, manually tagged NE corpus for Hungarian, which could serve as a reference corpus for training and testing NER systems. The *HunNer corpus* [Simon et al., 2006] project started as a consortial project with researchers from the University of Szeged, the Research Institute for Linguistics of Hungarian Academy of Sciences, and the Media Research Center of the Budapest University of Technology and Economics. The most important by-products of the project are the annotation guidelines based on the consensus of the project members. At the stage of corpus design, one of the primary factors was the compatibility with international standards, so the annotation schemes of CoNLL-2003

and LDC LCTL were adapted to Hungarian. The categories to be annotated are as follows: `Person`, `Organization`, `Location`, `Role` such as *elnök* ('President') and *szóvivő* ('spokesperson'), `Rank` such as *Sir* and *Lord*, `Brand/product`, `Title` of artworks, and `Miscellaneous`. Additionally, some metonymic names were also to be tagged: organization names referring to a location (`ORG:LOC`), and location names referring to an organization (`LOC:ORG`). This solution can be postulated as a kind of combination of Tag for Meaning and Tag for Tagging rules. The corpus itself stayed unfinished for reasons outside the author's control, but the annotation guidelines have proven to be remarkably durable.

Some parts of the annotation guidelines were used by researchers of the University of Szeged for building the *Criminal NE corpus*, which contains texts related to the topic of criminally liable financial offences. Articles were selected from the Heti Világgazdaság (HVG) subcorpus of the Hungarian National Corpus [Váradi, 2002]. The range of annotated NE categories was also based on the CoNLL-2003 annotation scheme, i.e. person, organization, location and miscellaneous names are tagged. The corpus has two annotated versions: one follows the Tag for Meaning rule, while the other one is annotated according to the standard Tag for Tagging approach.

The Szeged NER and Criminal NE corpora are freely available for research purposes[2], and the annotation guidelines of the HunNer corpus are also available[3].

4.2.4 The Size Factor

In support of the statement that gold standard NE tagged corpora are restricted in size, we take a closer look at the exact numbers in this subsection. Since the organizers of MUC-7 did not follow the standard train–devel–test set cut, counted optional fills allowed by the key (see MUC evaluation protocol, in Subsection 5.3.1), and use embedded annotations, we could not compile a table with exact figures about NE types in the datasets. CoNLL-2003 organizers, on the other hand, did provide such figures for the English data [Tjong Kim Sang and De Meulder, 2003].

Table 4.2 shows the number of NEs in the CoNLL-2003 data files, both by types (`LOC, MISC, ORG, PER`) and overall (*NEs* column). The number of tokens (*tokens*), and the proportion of NEs to the total number of tokens (*NE density*) are also listed. As can be seen in the table, location

[2] `http://www.inf.u-szeged.hu/rgai/nlp?lang=en&page=corpus_ne`
[3] http://krusovice.mokk.bme.hu/∼eszter/utmutato.pdf

	LOC	MISC	ORG	PER	NEs	tokens	density (%)
train	7,140	3,438	6,321	6,600	23,499	203,621	11.54
devel	1,837	922	1,341	1,842	5,942	51,362	11.57
test	1,668	702	1,661	1,617	5,648	46,435	12.16
total	10,645	5,062	9,323	10,059	35,089	301,418	11.64

Table 4.2: Number of NEs and tokens, and NE density per data file in CoNLL-2003 English data.

names are the most frequent in almost all data files and also in total, so good performance is expected in recognizing this category. NE density is higher in the test set than in the development set. We therefore expect that evaluating a system on the test set will lead to lower performance measures than those obtained using the development set. Since the MISC category is very diverse, and much smaller than other categories, it is reasonable to expect the lowest performance on this class. For all types except organizations, there are more instances in the development set than in the test set, so in the case of organization names we expect lower performance on the test set than on the development set. For actual results of our NER system, see Chapter 5 and 6.

	LOC	MISC	ORG	PER	NEs	tokens	density(%)
Szeged NER	1,501	2,041	20,433	1,921	25,896	225,963	11.46
Crimi T-f-M	5,049	1,917	8,782	8,101	23,849	562,822	4.24
Crimi T-f-T	5,391	854	9,480	8,121	23,846	562,822	4.24

Table 4.3: Number of NEs and tokens and NE density in Hungarian gold standard corpora.

Table 4.3 shows the number of NEs and tokens as well as NE density in Hungarian gold standard corpora. Here we do not give per data file numbers, since these corpora are not divided into train–devel–test sets by default. For comparable results, one should obtain the same cut as the one used when comparing our tool to another Hungarian NER system (for details, see Chapter 5). The first row of the table contains figures for the Szeged NER corpus, the second and the third for the two versions of the Criminal NE corpus (Tag for Meaning and Tag for Tagging).

As can be seen in Table 4.3, the number of organization names in the Szeged NER corpus is extraordinary high. NE density is similar to CoNLL, but most NEs are organization names. This can be attributed to the fact that texts are from business newswire articles, where names of firms, companies and institutions play an important role. There is great difference between the NE density of the Szeged NER corpus and the Criminal NE corpus, which can be a result of the difference in domain. In short business news, every sentence contains at least one, but often several NEs, while longer news articles of other topics are less crowded with NEs. Significant difference in NE density between the training and test set can cause a dramatic decrease in cross-evaluation results.

Another interesting question these figures raise is the change in the number of instances of NE types in the Criminal NE corpus when changing the annotation rule. The number of classes decreases when applying the Tag for Meaning rule, only the number of `MISC` increases. This is mainly caused by the fact that every article contains a header with the name of the newspaper, whose label changes from `ORG` to `MISC`, because it does not refer to an organization, but a newspaper in this context. Other metonymic shifts seem to be balanced between the classes.

4.3 Silver Standard Corpora

As illustrated above, gold standard datasets are highly domain-specific (mostly newswire) and are restricted in size. Researchers attempting to merge these datasets in the hope of acquiring a larger training corpus are faced with the problem of having to combine various tagsets and annotation schemes. Manual annotation of large amounts of text with linguistic information is a time-consuming, highly skilled, and delicate job, but large, accurately annotated corpora are essential for building robust supervised machine learning NER systems. Reducing the cost of annotation is therefore a key challenge.

There are several ways to reach this goal. One approach is to use semi-supervised or unsupervised methods, which do not require large amounts of labelled data (for details, see Section 5.3). Another approach is to generate resources automatically, or at least by applying NLP tools that are accurate enough to allow automatic annotation. Yet another approach is to use collaborative annotation and/or collaboratively constructed resources, such as Wikipedia, Wiktionary, Linked Open Data, or DBpedia. In the subsequent section we introduce a method which combines these approaches by automatically generating freely available NE tagged cor-

pora from Wikipedia.

The section[4] is structured as follows: in Subsection 4.3.1, we give an overview of related work. Subsection 4.3.2 contains a description of our method, and Subsection 4.3.3 shows how it is applied to Hungarian. The corpus format is described in Subsection 4.3.5. In Subsection 4.3.4, we present experiments and results on the newly generated datasets. Subsection 4.3.6 concludes the section with a summary.

4.3.1 Wikipedia and Named Entity Recognition

Wikipedia[5], a free multilingual Internet encyclopedia, written collaboratively by volunteers, is a goldmine of information: at the time of writing the article[6], Wikipedia contained about 21 million interlinked articles. Of these, 3,903,467 were in English, and 212,120 were in Hungarian[7]. Wikipedia has been applied for several NLP tasks such as WSD, ontology and thesaurus building, and question answering (see Medelyan et al. [2009] for a survey). It is recognized as one of the largest available collections of entities, and also as a resource that can improve the accuracy of NER. The most obvious utilization of Wikipedia for NER is extracting gazetteers containing person names, locations or organizations (e.g. Toral and Muñoz [2006]). Creating dictionaries of entities is also a common step of NE disambiguation [Bunescu and Pasca, 2006; Cucerzan, 2007]. Both supervised and unsupervised NER systems use such lists (e.g. Nadeau et al. [2006]). The knowledge embodied in Wikipedia may also be incorporated in NER learning as features, e.g. Kazama and Torisawa [2007] showed that automatic extraction of category labels from Wikipedia improves the accuracy of a supervised NE tagger.

Another approach to improve NER with Wikipedia is the automatic creation of training data. Richman and Schone [2008] built corpora for less commonly taught languages annotated with NE tags. They used the inherent category structure of Wikipedia to determine the NE type of a proposed entity. Nothman et al. [2008] used a similar method to create NE annotated text in English. They transformed Wikipedia links into NE annotations by classifying target articles into standard entity classes. Their

[4]This section is mainly based on our article [Simon and Nemeskey, 2012].

[5]`http://wikipedia.org`

[6]To preserve the coherence of results, we list the same figures as those published in our article. However, since our work is based on continuously changing resources, we also provide current figures in footnotes.

[7]Wikipedia at the time of writing the dissertation contains over 22 million articles, 4,110,000+ in English and 232,000+ in Hungarian.

approach to classification is based primarily on category head nouns and the opening sentences of articles where definitions are often given.

Our approach to the recognition and classification of NEs in corpora generated from Wikipedia involves mapping the DBpedia ontology classes to standard NE tags and assigning them to Wikipedia entities. Except for the Semantically Annotated Snapshot of the English Wikipedia (SASWP) [Atserias et al., 2008], no such automatically built corpora are freely available. SASWP provides a wide range of linguistic information: POS tags, dependency labels, WordNet supersenses and NE annotation according to BBN and CoNLL tagsets. Even though the SASWP NEs were tagged by the best open source taggers available, the tags provided here, based on the manual judgement of thousands of Wikipedia volunteers, are more reliable.

Given the huge number of Wikipedia articles, we can build sufficiently large corpora for less resourced languages as well, as our method is largely language-independent. We demonstrate this on Hungarian. There are manually annotated gold standard datasets for Hungarian, as described above, but the one presented here is the first automatically NE annotated corpus for Hungarian.

4.3.2 Creating the English Corpus

Our goal was to create a large NE annotated corpus, automatically generated from Wikipedia articles. We followed a similar path to Nothman et al. [2008] and broke down the process into four steps:

1. Classifying Wikipedia articles into NE classes.
2. Parsing Wikipedia and splitting articles into sentences.
3. Labelling NEs in the text.
4. Selecting the sentences for inclusion in the corpus.

In this subsection, we describe how these steps were implemented, explain the general approach and its execution for English. Subsection 4.3.3 describes how the idea was adapted to Hungarian.

Articles as entities

Many authors such as Kazama and Torisawa [2007] and Nothman et al. [2008] used semi-supervised methods based on Wikipedia categories and

text to classify articles into NE types. To avoid the inevitable classification errors, we obtain entity type information from the DBpedia knowledge base [Bizer et al., 2009], which presents type, properties, home pages, and other information about pages in Wikipedia in a structured form. DBpedia supplies us with high precision information about entity types at the expense of recall, since only a third of English Wikipedia pages are covered by DBpedia at the time of writing[8].

The types in DBpedia are organized into a class hierarchy, available as an OWL[9] ontology containing 319 frequent entity categories[10], arranged into a taxonomy under the base class `owl:Thing`. Most classes belong to one of the 6 largest sub-hierarchies: `Person`, `Organization`, `Event`, `Place`, `Species` and `Work`. The taxonomy is rather flat: the top level contains 44 classes, and there are several nodes with a branching factor of 20.

Entity types are extracted automatically from Wikipedia categories. However, the mapping between Wikipedia categories and classes in the DBpedia ontology is manually defined. This, together with the fact that the existence of the reference ontology prevents the proliferation of categories observable in Wikipedia, ensures that type information in DBpedia can be considered gold quality.

From NER annotation standards available we chose to use the CoNLL-2003 NE types. It is not difficult to see the parallels between the DBpedia sub-hierarchies `Person`, `Organization` and `Place` and the corresponding CoNLL NE types. The fourth category, `MISC` is more elusive; according to the CoNLL annotation guidelines, the sub-hierarchies `Event` and `Work` belong to this category, as well as various other classes outside the main hierarchies.

While the correspondence described above holds for most classes in the sub-hierarchies, there are some exceptions. For instance, the class `SportsLeague` is part of the `Organization` sub-hierarchy, but according to the CoNLL annotation scheme, it should be tagged as `MISC`. To avoid misclassification, we created a file of DBpedia class–NE category mapping. Whenever an entity is evaluated, we look up its class as well as the ancestors of its class, and assign the category of the class that matches the entity most closely. If no match is found, the entity is tagged with `O`, i.e. it is not a NE. Since we take advantage of the inheritance hierarchy, the mapping list remains short: it contains only the root classes of the main hi-

[8]Indeed, the number of DBpedia entries is growing with each new release. Currently, 2.35 million entities are classified in the ontology, thus more than half of the total number.

[9]`http://www.w3.org/TR/owl-ref/`

[10]Currently 359 classes.

erarchies, exceptions like those mentioned above, and the various classes that belong to the `MISC` category according to CoNLL annotation guidelines.

As of version 3.7[11], the DBpedia ontology allows multiple inheritence, i.e. classes can have more than one superclasses, resulting in a directed acyclic graph. Since selecting the right superclass and hence, the right CoNLL tag, for classes with more than one parent cannot be reliably done automatically, the class–category mapping had to be determined manually. The only such class in version 3.7, `Library` can be traced back to both `Place` and `Organization`; its CoNLL tag is `LOC`. Using the mapping, we compiled a list that contains all entities in DBpedia tagged with the appropriate CoNLL category (see Table 4.4).

DBpedia	**CoNLL**	**DBpedia**	**CoNLL**
Person	PER	Library	LOC
Place	LOC	MeanOfTransportation	MISC
Organization	ORG	ProgrammingLanguage	MISC
Award	MISC	Project	MISC
EthnicGroup	MISC	SportsLeague	MISC
Event	MISC	SportsTeamSeason	O
Holiday	MISC	Weapon	MISC
Ideology	MISC	Work	MISC
Language	MISC	PeriodicalLiterature	ORG

Table 4.4: Mapping between DBpedia entities and CoNLL categories.

We note here that our method can be trivially modified to work with any tagset compatible with the DBpedia ontology. Indeed, the DBpedia classes themselves define a NE tagset, which allows for a more fine-grained NE type hierarchy.

Parsing Wikipedia

Wikipedia is a rich source of information: in addition to the article text, a large amount of data is embedded in infoboxes, templates, and the category and link structures. For the current task, we only extracted links between articles and article text. In addition to in-article links, our method

[11]Since then, version 3.8 was released.

takes advantage of the redirect and interlanguage links. The English corpus is based on the Wikipedia snapshot as of January 15, 2011. The XML files were parsed by the `mwlib` parser[12]; the raw text was tokenized by a modified version of the `Punkt` sentence and word tokenizers [Kiss and Strunk, 2006]. For lemmatization, we used the `WordNet Lemmatizer` in NLTK[13], and for POS tagging, the `hunpos` tagger [Halácsy et al., 2007].

Named Entity labelling

In order to automatically prepare sentences where NEs are accurately tagged, two tasks need to be performed: identifying entities in the sentence and assigning the correct tag to them. Sentences for which accurate tagging could not be accomplished must be removed from the corpus. Our approach is based on the work of Nothman et al. [2008]. Wikipedia cross-references found in the article text are used to identify entities. We assume that individual Wikipedia articles describe NEs, so a link to an article can then be perceived as a mapping that identifies its anchor text with a particular NE.

The discovered entities are tagged with the CoNLL label assigned to them in the entity list extracted from DBpedia. If the link target is not in the entity list, or the link points to a disambiguation page, we cannot determine the type of the entity, and tag it as `UNK` for subsequent removal from the corpus. Links to redirect pages are resolved to point instead to the redirect target, after which they are handled as regular cross-references. Finally, sentences with `UNK` links in them are removed from the corpus.

Strictly speaking, our original assumption of equating Wikipedia articles with NEs is not valid: many pages describe common nouns (e.g. *Book*, *Aircraft*), calendar-related concepts (e.g. *March 15, 2007*), or other concepts that fall outside the scope of NER. To increase sentence coverage, we modified the algorithm to prevent it from misclassifying links to these pages as unknown entities and discarding the sentence. The list of non-entity links and the way of handling them is as follows:

Common noun links are filtered by POS tags: if they do not contain `NNP`, they are ignored.

Time expression links require special attention, because dates and months are often linked to the respective Wikipedia pages. We circumvented this problem by compiling a list of calendar-related

[12] `http://code.pediapress.com`

[13] `http://nltk.org/_modules/nltk/stem/wordnet.html#WordNetLemmatizer`

pages and adding them to the main entity list tagged with the CoNLL category O.

Lower case links for entities referred to by common nouns, such as *republic* to *Roman Republic* are not considered NEs and are ignored.

In a Wikipedia article, typically only the first occurrence of a particular entity is linked to the corresponding page. Subsequent mentions are unmarked and often incomplete; for example, family names are used instead of full names. To account for such mentions, we apply Nothman et al.'s [Nothman et al., 2008] solution. For each page, we maintain a list of entities discovered in the page so far and try to associate capitalized words in the article text with these entities. We augment the list with the aliases of every entity, such as titles of redirect pages that target it, the first and last names in the case of person names, and any numbers in the name. If the current page is a NE, the title and its aliases are added to the list as well; moreover, as Wikipedia usually includes the original name of foreign entities in the article text, localized versions of the title are also added to the list as aliases. Nothman et al. used a trie to store the entity list, while we use a set. We also use a larger number of alias types.

Additionally, there are some special cases to our method, which are detailed below.

Derived words. According to CoNLL guidelines, words derived from NEs are tagged as MISC. We complied with this rule by tagging as MISC each entity whose head is not a noun, as well as those where the link's anchor text is not contained in the entity's name. The most prominent example for such entities are nationalities, which can be linked to their home country, a LOC; e.g. *Turkish* to *Turkey*. Our solution assigns the correct tag to these entities.

First word in a sentence. As first words are always capitalized, labelling them is difficult if they are unlinked and not contained in the entity alias set. In these cases, the decision is based on the POS tag of the first word: if it is NNP, we tag it as UNK; otherwise as O.

Reference cleansing. Page titles and anchor texts may contain more than just a name. For example, personal titles are part of the page titles in Wikipedia, but they are not considered NEs according to the CoNLL annotation scheme. To handle personal titles, we extracted a list from the Wikipedia page *List of titles*, which contains titles in many languages. We removed manually all titles that also function as given

names, such as *Regina*. If a link to a `Person` or `UNK` entity, or an unlinked entity starts with, or consists solely of a title in the list, we tag the words that make up the title as `O`.

Punctuation marks. Around names they may become part of the link by mistake. We tag all punctuation marks after the name as `O`.

Discarding sentences

As mentioned above, sentences with words tagged as `UNK` are discarded. Furthermore, there are many incomplete sentences in Wikipedia text: image captions, enumeration items, contents of table cells, etc. On the one hand, these sentence fragments may be of too low quality to be of any use in the traditional NER task. On the other hand, they could prove to be invaluable when training a NE tagger for user generated content, which is known to be noisy and fragmented. As a compromise, we included these fragments in the corpus, but labelled them as "low quality", so that users of the corpus can decide whether they want to use them or not. A sentence is labelled as such if it either lacks a punctuation mark at the end, or contains no finite verb.

4.3.3 Creating the Hungarian Corpus

The procedure described in the previous subsection was used to generate the Hungarian corpus as well. However, typological differences posed several problems. In this subsection, we describe the changes in the method prompted by differences between the two languages related to the labelling of NEs.

Parsing the Hungarian Wikipedia

The Hungarian corpus is based on the Wikipedia snapshot as of March 9, 2012. Similarly to the English corpus, the XML files were parsed by the `mwlib` parser. For tokenization and sentence segmentation, we used an in-house statistical tool tailored for Hungarian, which has been trained on the Szeged corpus [Csendes et al., 2004] and handles the peculiarities of Hungarian orthography, such as the periods placed after numbers in date expressions. Lemmatization and morphological analysis were performed by `hunmorph` [Trón et al., 2005a], and `hundisambig` [Halácsy et al., 2005] was used to select the correct analysis based on context. `Hunmorph` outputs KR codes [Rebrus et al., 2012], which, in addition to the POS category,

also include inflectional information, making it much better suited to agglutinative languages than Penn Treebank POS tags [Marcus et al., 1993]. One shortcoming of the KR code is that it does not differentiate between common and proper nouns. Since in Hungarian only proper nouns are capitalized, we can usually decide whether a noun is a proper noun based on the initial letter. However, this rule cannot be used if the noun is at the beginning of a sentence, so sentences that begin with unidentified nouns have been removed from the corpus.

Named Entity labelling in Hungarian

For well-resourced languages, DBpedia has internationalized chapters, but not for Hungarian. Instead, the Hungarian entity list comprises of pages in the English list that have their equivalents in the Hungarian Wikipedia. Two consequences follow.

First, in order to identify which pages denote entities in the Hungarian Wikipedia, an additional step is required, in which Hungarian equivalents of English pages are added to the entity list. English titles are retained because (due to the medium size of the Hungarian Wikipedia) in-article links sometimes point to English articles.

Second, entities without a page in the English Wikipedia are absent from the entity list. This gives rise to two potential problems. One is that compared to English, the list is relatively shorter: the entity per page ratio is only 12.12%, as opposed to the 37.66% of the English Wikipedia. The other issue is that, since missing entities are mostly Hungarian people, places and organizations, a NE tagger that takes the surface form of words into account might be misled as to the language model of entity names. To overcome these problems, the list has to be extended with Hungarian entity pages that do not have a corresponding English page. This is left for future work.

To annotate the Hungarian corpus with NE tags, we chose to follow the annotation scheme of the Szeged NER corpus, because it is similar to the CoNLL standard, which was used for the English Wikipedia corpus. There are some categories which are not considered NEs in Hungarian (see Subsection 2.3.1). We therefore modified the mapping from DBpedia categories to NE labels used when creating the English corpus. The entity types in Table 4.4 whose labelling was changed from `MISC` to `O` are: ethnic group, event, holiday, ideology, and language.

There is another special case in Hungarian: NEs can be subject to compounding, and, unlike in English, the common noun following the NE is joined with a hyphen, so they constitute one token. The joint com-

mon noun can modify the original reference of the NE, depending on the meaning of the common noun. For example, in the compound *Nobel-díj* ('Nobel Prize'), the common noun changes the labelling from `PER` to `MISC`, while in the case of the compound *WorldCom-botrány* ('WorldCom scandal'), the NE tag changes from `ORG` to `O`. Additionally, inflections of acronyms and foreign names ending with a non-pronounced vowel have similar surface form to the aforementioned compounds, e.g. *MTI-t* ('MTI.ACC'), *Shakespeare-rel* ('with Shakespeare'). It is important to distinguish these types of hyphenated NEs, because inflections do not change NE labelling in this case, in contrast to some types of compounds. Zsibrita et al. [2010] use a quite simple method based on morphological codes and relative lemmas to distinguish hyphenated NE compounds from inflected NEs. This solution may be built in our system in the future.

Error analysis

The automatic annotation of the Hungarian Wikipedia corpus was manually checked on a sample corpus[14]. Of the whole corpus containing 19 million tokens, sentences of 18,830 tokens were randomly selected for inclusion in the sample corpus. This was annotated by hand, then the labels given by us were compared to the labels emitted by the automatic method. If the automatic tagging method is considered an annotator, the F-measure can be considered a kind of inter-annotator agreement. Results are shown in Table 4.5.

	precision(%)	recall(%)	$F_{\beta=1}$(%)	NEs(#)
`LOC`	98.72	95.65	97.16	161
`MISC`	95.24	76.92	85.11	26
`ORG`	89.66	89.66	89.66	29
`PER`	88.30	89.25	88.77	93
total	94.33%	91.59%	92.94	309

Table 4.5: Results of manual evaluation on the sample corpus.

The confusion matrix for the four categories (Table 4.6) shows that misclassification is quite rare. For measuring the inter-annotator agreement in another way, we also counted the Cohen's κ from these scores, which is resulted in a 0.967 value. Since the strength of agreement is said to be perfect

[14]This subsubsection is based on our article [Nemeskey and Simon, 2012].

above 0.8 κ value according to Landis and Koch [1977], we can conclude that the annotation of our automatically generated Hungarian Wikipedia corpus reaches the gold standard quality. However, if we compare the 92.94% overall F-measure to the 99.6% agreement rate achieved by human annotators of the Szeged NER corpus [Szarvas et al., 2006a], our result seems to be quite low. We assume that more accurate tagging will require investigating the error types and correcting the method.

Auto↓ / Gold→	PER	ORG	LOC	MISC
PER	83	1		2
ORG		26	1	1
LOC		1	154	
MISC			1	20

Table 4.6: The confusion matrix of the manually annotated sample corpus.

Misclassification can be explained by two main reasons. In the first case, the category information of an entity in the DBpedia is incorrect. For example, the ontology class of *Magyar Tudományos Akadémia* ('Hungarian Academy of Sciences') in the DBpedia is `WorldHeritageSite`, which causes that the label `LOC` is assigned to it, instead of the right choice, `ORG`. Similarly, if only one reference of a referentially multivalent name is included in the DBpedia ontology, the same label will be assigned to the name in every context, irrespectively of its actual usage. Second, misclassification can be caused by the fact that some in-article links in Wikipedia may not point to the correct page. For example, the editor of a Hungarian article made a link from a part of the company name 'Walt Disney Co.' to the page of the person Walt Disney, therefore several versions of this name (*Disney*, *Walt Disney* etc.) mentioned in the article are always labelled as a person name.

Other error types of NER, such as identifying a non-NE as a NE (false positive), not recognizing a NE (false negative), or not finding the correct boundaries of a NE are more frequent. The figures of these error types are shown in Table 4.7.

The most frequent reason for missing the correct boundaries of a NE is that page titles and anchor texts may contain more than just a name. These extra elements of a link usually are personal titles, which are handled in the English corpus, but not yet in the Hungarian one. A manually collected list of titles, such as *király* or *pápa*, can be used as a stopword list in the future.

	PER	ORG	LOC	MISC
False positive	1	0	1	0
False negative	3	0	5	4
Incorrect boundaries	7	1	0	0

Table 4.7: Other error types in the sample corpus.

Moreover, the explanatory elements in Wikipedia page titles sometimes inhibit the recognition of a whole NE. This occurs in such cases where the title of the linked article is not a proper name, but contains a proper name already identified by the method, e.g. *Ókori Róma* ('Ancient Rome'), *Magyar Wikipédia* ('Hungarian Wikipedia'). Since these page titles are not contained by DBpedia ontology classes which are considered NEs, they remain unlabelled.

All of the false negative names in the `MISC` class are entries in bibliographical lists of authors' works. Since these titles do not have their own Wikipedia articles, i.e. they are not linked to a page, their recognition is not possible by our method. Moreover, titles of artworks can contain any kind of linguistic units, so even by applying all of our filtering techniques we cannot discard sentences containing such NEs. Since the recognition and processing of bibliographical references is a full-fledged NLP task, within this workflow we cannot accomplish it.

A further type of error is caused by mistakes of the applied text preprocessing tools. For example, if the sentence splitter did not recognize period as a part of the abbreviation (e.g. *Warner Bros.*), but as a sentence ending punctuation mark, it will not be annotated within the boundaries of the name. If a period is inside a link and is interpreted as a sentence end because of the sentence splitter's overgeneralization tearing the link apart, the NE will not be labelled properly. The initial word of a sentence is considered a potential NE only if it is identified as a noun. Thus, some sentence-initial NEs can remain unlabelled because of the tagging errors of the morphosyntactic disambiguator. For example, in the sentence *Hél visszaengedte volna* ('Hel would have allowed him to return'), the word 'Hél' was identified as a verb by `hundisambig`, so it was not considered a potential NE. These and similar problems may be solved by improving the performance of the pre-processing tools or applying other ones.

Correcting the aforementioned error types and several other ones caused by the deficiencies of our method is left for future work. Our preliminary results, however, show that after error correction we will get gold

standard quality corpora for training and testing NER systems.

4.3.4 Evaluation

Automatically generated corpora can be very useful for improving NER in several ways: (a) for less resourced languages, they can serve as training corpora *in lieu* of gold standard datasets; (b) they can serve as supplementary or independent training sets for domains differing from newswire; (c) they can be sources of large entity lists, and (d) feature extraction.

To evaluate our corpora we used the `hunner` system [Varga and Simon, 2007], which was originally developed for labelling NEs in Hungarian texts, but can be tuned for different languages as well (for details see Subsection 5.3.2). Corpus-specific features (e.g. chunks, Wikipedia links) were removed to achieve better comparability, so the feature set consists of gazetteer features, sentence start and end position, Boolean-valued orthographic properties of the word form, string-valued surface properties of the word form, and morphological information.

We used the CoNLL standard method for evaluation, calculating precision and recall values, and F-measure. For more details on evaluation metrics, see Subsection 5.3.1.

Wikipedia data

Our automatic annotation process retains all Wikipedia sentences which remained after discarding the sentences containing unknown NEs and low quality sentences, so sentences without NEs are also included in the corpus. The rationale behind this is that we wanted to preserve the original distribution of names in Wikipedia as much as possible. However, after further investigation of the NE density in our corpora and the gold standard corpora, we decided not to include the sentences without NEs in evaluation datasets, thus getting more comparable corpora.

	enwiki	**enwiki filtered**	**CoNLL**
tokens	60,520,819	21,718,854	301,418
NEs	3,169,863	3,169,863	35,089
NE density (%)	5.23	14.59	11.64

Table 4.8: Corpus size and NE density of the English Wikipedia corpus compared to the CoNLL-2003 gold standard dataset.

	huwiki	huwiki filtered	Szeged NER
tokens	19,108,027	3,512,249	225,963
NEs	456,281	456,281	25,896
NE density (%)	2.38	12.99	11.46

Table 4.9: Corpus size and NE density of the Hungarian Wikipedia corpus compared to the Szeged NER corpus.

Tables 4.8 and 4.9 summarize the data regarding corpus size and NE density. The English (*enwiki* column) and the Hungarian Wikipedia (*huwiki*) corpora originally have NE densities of 5.23% and 2.38%, respectively. In comparison to the gold standard datasets (CoNLL, Szeged NER) these counts are quite low. This can be due to the difference between domains: newswire articles usually contain more NEs. The other reason might be that we discarded sentences containing unidentified NEs (cf. Subsection 4.3.2). After filtering out sentences which do not contain NEs, NE density of our newly generated corpora became quite similar to that of gold standard datasets (*enwiki filtered*, *huwiki filtered*).

Experiments and results

The English Wikipedia corpus was evaluated against itself and the CoNLL-2003 corpus. Since the filtered English Wikipedia corpus containing only the sentences with NEs is still very large, our experiments were performed with a sample of 3.5 million tokens, the size of our filtered Hungarian corpus, divided into train and test sets (90%-10%).

As discussed in Subsection 4.2.2, training and testing across different corpora decreases F-measure. The situation here is similar (see Table 4.10 for English results): when tested against the CoNLL test set, performance of the NE tagger trained on Wikipedia (*enwiki–CoNLL*) is lower than that achieved by training a model on the CoNLL training set (*CoNLL–CoNLL*), and the same is true for the other direction (*CoNLL–enwiki*).

We also made experiments demonstrating that Wikipedia-derived corpora can also be used for improving NER accuracy in other ways. First, we collected gazetteer lists from the corpus for each NE category, which improved the overall F-measure when used by the NE tagger for training and testing on the CoNLL datasets (*CoNLL with wikilists*). Second, the CoNLL datasets were labelled by the model trained on the Wikipedia corpus, then we used these labels as extra features when training the system

train	test	precision	recall	F-measure
CoNLL	CoNLL	85.13	85.13	85.13
enwiki	enwiki	72.46	73.33	72.89
enwiki	CoNLL	56.55	49.77	52.94
CoNLL	enwiki	48.19	56.07	51.83
CoNLL with wikilists	CoNLL	86.33	86.35	86.34
CoNLL with wikitags	CoNLL	85.88	85.94	85.91

Table 4.10: Results for the English Wikipedia corpus.

on the CoNLL train set (*CoNLL with wikitags*). Both methods result in improved F-measure on the CoNLL test set.

Since in Hungarian NE tagging we followed the Szeged NER corpus annotation scheme, we performed experiments on this dataset. Hungarian results are similar to the English ones (see Table 4.11), the only difference is that F-measures for Hungarian are significantly higher. This can be due to the fact that the `MISC` category for Hungarian contains less types of names, thus the inconsistency of this class is smaller. In contrast to the CoNLL corpus, the Szeged NER corpus was accurately annotated with an inter-annotator agreement over 99%, which can also be a source of higher F-measures.

train	test	precision	recall	F-measure
Szeged	Szeged	94.50	94.35	94.43
huwiki	huwiki	90.64	88.91	89.76
huwiki	Szeged	63.08	70.46	66.57
Szeged	huwiki	64.01	51.60	57.13
Szeged with wikilists	Szeged	95.48	95.48	95.48
Szeged with wikitags	Szeged	95.38	94.92	95.15

Table 4.11: Results for the Hungarian Wikipedia corpus.

Due to the quite good F-measure resulting from training on the Hungarian Wikipedia corpus and testing on the corresponding test set, we can say that our Hungarian Wikipedia corpus can serve as a training corpus to build NE taggers for non-newswire domains.

4.3.5 Data Description

Our corpora are available under the Creative Commons Attribution-Share-alike 3.0 Unported License (CC-BY-SA), the same license under which the text of Wikipedia is released. Data files can be freely downloaded from `http://hlt.sztaki.hu/resources/hunnerwiki.html`. The corpora are also distributed through the META-SHARE network[15], which is an open, distributed facility for exchanging and sharing resources, and is one of the lines of action of META-NET, a Network of Excellence funded by the European Commission.

The files are in multitag format. Content lines are tab separated; there is one column for the tokens plus one column per tagset. Sentence boundaries are marked by empty lines. The linguistic tags include the lemmatized form of the word and its POS tag. Two NE tags are included with each word: the most specific DBpedia category it belongs to and the CoNLL NE tag. While NE tags can be considered silver standard, linguistic tags are provided on a best-effort basis.

4.3.6 Summary

We have presented freely available NE tagged corpora for English and Hungarian, fully automatically generated from Wikipedia. In contrast to the methods used so far for automatic annotation of NEs in Wikipedia texts, we applied a new approach, mapping DBpedia ontology classes to standard CoNLL NE tags and assigning them to Wikipedia entities. Following Nothman et al. [2008], the process can be divided into four main steps: classifying Wikipedia articles into NE classes, parsing Wikipedia and splitting articles into sentences, labelling NEs in the text, and selecting sentences for inclusion in the corpus.

The large amount of Wikipedia articles opens the possibility of building large enough corpora for otherwise less resourced languages such as Hungarian. Due to the properties of Hungarian, some steps are slightly different, and special linguistic phenomena pose several problems related to the NER task at hand.

Automatically generated corpora can be useful for improving NER in several ways. We showed that using gazetteer lists extracted from our corpora and extra features supplied by the model trained on our corpora both improve F-measure. Moreover, our Hungarian corpus can serve as a training corpus for more general domains than the classic newswire.

[15] `http://www.meta-share.eu`

4.4 Conclusion

Several current trends concerning the NER task emerge from our overview. The main efforts are directed to reducing the annotation labour, robust performance across domains, and scaling up to fine-grained entity types.

Machine learning algorithms typically learn their parameters from a corpus, and systems are evaluated by comparing their output to another part of the corpus or to another corpus. Thus, for the purpose of developing NER systems, corpora containing rich linguistic information are required. Datasets manually enriched with annotation are called gold standard corpora. They have to meet several requirements, so building such corpora is a time-consuming, delicate job, which requires large amounts of resources. Thus, reducing the annotation cost is one of the main trends in NLP in general, and NER in particular.

Second, large and accurately annotated corpora are essential for building robust supervised machine learning NER systems. The gold standard datasets currently available are highly domain-specific and restricted in size. Experiments confirmed that cross-domain evaluation of NER systems results in low F-measure. Thus, current efforts are directed to reach robust performance across domains.

The third trend in NER research is scaling up to fine-grained entity types. Classic gold standard datasets use coarse-grained NE hierarchies, taking into account only the three main classes of names (`PER`, `ORG`, `LOC`) and certain other types depending on the applied annotation scheme (e.g. `MISC` in CoNLL, and time and numerical expressions in MUC). Fine-grained NE hierarchies also exist [Sekine et al., 2002; Weischedel and Brunstein, 2005; ACE, 2008], but when used for evaluation, they have to be mapped to the classic coarse-grained typology, which is far from trivial.

In this chapter, we presented a new method to achieve at least a few of these goals. Building automatically generated corpora from collaboratively constructed resources such as Wikipedia and DBpedia significantly decreases annotation labour. While continuous manual annotation is not feasible for building large corpora, our method can be used for generating even larger datasets, thus automatically generated corpora are not restricted in size. Using continuously growing collaboratively constructed resources also creates the possibility of building corpora with a more fine-grained NE hierarchy than one consisting of the classic NE types. However, reaching robust performance across domains still remains a problem and needs further investigation.

Chapter 5

Approaches to Named Entity Recognition

The two main approaches to NER as well as to other NLP tasks are using rationalist or empiricist methods. Section 5.1 gives a brief introduction to these approaches, and shows where they stand in philosophy, in linguistics and in NLP. Sections 5.2 and 5.3 focus on NER: they introduce the rationalist and the empiricist methods, respectively, which are generally used for the task, and include descriptions of a rule-based and a statistical NER system for Hungarian, to the development of which we contributed. Finally, in Section 5.4, we conclude with a summary of advantages and disadvantages of the two approaches.

5.1 Rationalist and Empiricist Approaches

One of the biggest challenges in NLP is providing computers with sophisticated knowledge for being able to process language. There are several approaches to reach this goal, and they can be divided into two main groups. One of them is the rationalist approach, which uses rules written by a linguist, thus providing the computer with linguistic information explicitly. The second one is the empiricist methodology, where the computational linguist gives text resources to the computer, which in turn uses them to teach itself.

This dichotomy is also valid for linguistics and the cognitive sciences, and has its roots in *philosophy*. It is historically related back to early debates about rationalism versus empiricism in the 17th century. Descartes and Leibniz took the rationalist position, asserting that all truth has its origins in human thought and in the existence of innate ideas implanted in our

minds from birth. The source of innate ideas is God, so the source of all knowledge is divine revelation. In contrast, other philosophers such as Locke argued that sensory experience has priority over revelation. They took the empiricist view, and said that our primary source of knowledge is the experience of our faculties.

In the context of *linguistics*, this debate leads to the following question: to what extent does human linguistic experience, versus our innate language faculty, provide the basis for our knowledge of language? Chomsky's work in general and, more specifically, his views on language acquisition are very much in the rationalist camp. Chomsky argues that it is difficult to see how children can learn something as complex as the natural language from the limited input they hear, so the language faculty must be innate. This is often called the problem of the poverty of stimulus. On the other side the behaviorists agree with Locke that the mind at birth is a *tabula rasa*, and language is entirely learned, are clearly empiricists. In fact, the question of language acquisition cannot be simplified to such a dichotomy, but is driven by many factors [Harley, 2001].

Abney [1996] shows that arguments for statistical methods in linguistics also come from the area of language acquisition. Experimental evidence shows that children do not acquire their first language without errors, and that these errors are not necessarily arbitrary but may clearly follow rules or patterns. This suggests that at each stage of development the child entertains different, sometimes erroneous hypothesis grammars [Serény et al., 2009]. Changes in child grammar are actually reflected in changes in relative frequencies of structures: children experiment with rules for certain periods of time. During the trial period, both the new and old versions of a rule co-exist, and the probability of using one or the other changes with time, until the probability of using the old rule finally drops to zero. Thus, the child's grammar is a probabilistic grammar.

In *NLP*, this issue surfaces in debates about the priority of corpus data versus linguistic introspection in the construction of computational models. The rationalist approach is often described as rule-based, since linguists following this approach create rules based on introspection. Introspection is a little informal psycholinguistic experiment performed by linguists on themselves with such questions as "Can you say this?" or "Does this mean this?". The answers for these questions can be postulated as exact linguistic data only if one accepts the assumption that humans innately have knowledge of language. On the opposing side, statistical or data-driven approaches obtain linguistic knowledge from vast collections of concrete example texts, i.e. corpora (cf. Chapter 4). The machine learning algorithm then learns patterns of language units and linguistic

phenomena, depending on the application.

5.1.1 The Two Camps in the 20th Century

In this subsection, we give a brief overview of the history of NLP. We only mention the main steps and findings of the followers of the two camps. For a more detailed description, we direct the reader to the essential introductory work of Jurafsky and Martin [2000] and to Brill and Mooney's article about NLP [Brill and Mooney, 1997].

The history of NLP dates back to the period of World War II, when several military purpose developments gave great impetus to NLP research. One of the main goals was decoding encrypted messages sent by enemies, a task which can be postulated as the roots of MT. Later, Shannon's *noisy channel* model [Shannon, 1948] was applied to human language: several NLP tasks can be resolved if they are treated as a decoding problem in a noisy channel. MT can also be considered a noisy channel problem: we consider a string of the source language as an observation of the target language version that has been sent through a noisy channel. The task of the decoder is then to find the original string of the target language. Since then, this approach had proven highly successful mostly in speech recognition, but also in other NLP tasks such as spellchecking [Brill and Moore, 2000] and text normalization [Oravecz et al., 2010].

Shannon's other invention was borrowing the concept of *entropy* from thermodynamics and applying it to the measurement of information capacity of a channel. This was one of the fundamental steps of information theory. The concept of entropy was later applied for the information content of a language, and in 1951, Shannon performed the first calculations of entropy for English using probabilistic techniques [Shannon, 1951].

In the 1950s, behaviorism was thriving in psychology, while within linguistics, the main insight was to use distributional information, i.e. the environment a word can appear in, as the tool for language study (e.g. Harris [1951]). In 1957, *Chomsky* published his famous work *Syntactic structures* [Chomsky, 1957], and in 1959, his review on Skinner's *Verbal Behavior* [Chomsky, 1959]. These works redefined the goals of linguistics dramatically: linguistics should not be merely descriptive, but should be concerned with the question of how children acquire the language, and what the common, universal properties of human language are. According to Chomsky's point of view, these phenomena cannot be studied through data, using "shallow" corpus-based methods.

Chomsky's arguments were very influential: much of the work on

corpus-based language learning was halted. Researchers in artificial intelligence and NLP adopted this rationalist approach and used rule-based representations of grammars and knowledge until the 1980s.

Rule-based systems have many *disadvantages*, however. The number of possible parses of a sentence within the generative paradigm can be huge, snowballing as sentences grow longer [Abney, 1996]. Parsing sentences with such a large number of potential parses was not feasible computationally in the 1980s, and it is not effective even in the 21st century.

A remarkable property of human language comprehension is its error tolerance, which is not mirrored in rationalist methods. For example, the sentence 'Thanks for all you help' has one grammatical analysis: *thanks for all those who you help*, which would be assigned to this sentence by a rule-based system. However, it is preferably interpreted as an erroneous version of *thanks for all your help*. Since the latter analysis is more frequent, a statistical method using frequency information would perform better on analysing this sentence. Human language texts are crowded with similar errors, so processing them requires more robust solutions than rule-based systems can provide.

Developing rule-based systems remained difficult, requiring a great deal of domain-specific knowledge engineering. In addition, the systems were brittle and not interchangable across different domains and tasks. Partially in reaction to these problems, in the late 1980s and in the 1990s, focus has shifted from rationalist to empirical methods.

In the meantime, the first computer-readable *corpus*, the Brown Corpus [Kucera and Francis, 1967] was created in the US, which then inspired a whole family of corpora, including the Lancester-Oslo-Bergen Corpus [Leech et al., 1983], Brown's British English counterpart, and the London-Lund Corpus [Svartvik, 1990]. These constitute the first generation of corpora, which have been especially influential in the development of English corpus linguistics.

In the 1980s, the *stochastic* paradigm played a huge role in the development of speech recognition algorithms. Speech researchers were quite successful using models based on the noisy channel metaphor and Hidden Markov Models (HMMs) that vastly overperformed the previous knowledge-based approaches. The success of statistical methods in speech then spread to other areas of NLP, first to POS tagging [Bahl and Mercer, 1976], which can be now performed at an accuracy close to human performance (it is usually said to be more than 95%).

The 1990s are often called the period of the *return of empiricism*. Probabilistic and data-driven models had become standard throughout the entire field of NLP, mainly due to their robustness and extensibility. Unlike

rule-based methods, statistical methods can produce a probability estimate for each analysis, thereby ranking all possible alternatives. This is a more flexible approach, which can improve robustness by allowing the selection of a preferred analysis even when the underlying model is inadequate (cf. the example sentence above 'Thanks for all you help').

Statistical approaches also have disadvantages. For training and testing, they require large amounts of accurately annotated data, as illustrated in Chapter 4. Thus, manual labour has not been removed from NLP, but shifted to other areas. In addition, there are certain tasks which are far from being resolved, even by statistical approaches. This is the situation, for example, with MT, where researchers have high hopes for statistical methods, but the output of such systems is still far from ideal.

Indeed, rule-based systems also have their advantages. One of them is that experts have greater control over the actual language processing. This makes it possible to systematically correct mistakes in the software and give detailed feedback to the user, especially when rule-based systems are used for language learning.

As the strengths and weaknesses of statistical and rule-based systems tend to be complementary, current research attempts to deal with *hybrid* solutions that combine the two methodologies.

5.2 Rule-based Systems

NER systems have the ability to recognize and classify previously unknown entities, based on rules triggered by distinctive features associated with positive and negative examples. While early studies mostly made use of hand-crafted rules, more recent ones use supervised machine learning. This is demonstrated by the fact that only three systems out of eight were statistical in MUC-7 [Chinchor, 1998b], while 16 machine learning systems were presented at CoNLL-2003 [Tjong Kim Sang and De Meulder, 2003], a forum devoted to learning techniques.

NER requires specialized linguistic knowledge about the structure or composition of each type of name. For example, person names usually consist of first names and last names, with optional name prefixes and suffixes, and many organization names contain acronyms such as *Corp.* or *Ltd.* Following McDonald's terminology [McDonald, 1996], we call these *internal evidence*, derived from within the sequence of words that comprise the name. However, there are many names that do not provide the structural indication of their category membership. For example, most country names, such as *France*, have no internal structure indicating that the name

is a country.

Thus, for recognizing and classifying names, knowledge about how names appear in free text is also required. This knowledge consists of contextual clues about how each type of name may appear. For example, person names may have professional titles or descriptions preceding or following the name. These are examples of *external evidence*. Language units cannot be categorized based on internal evidence alone and require external evidence from the context as well.

For this reason, a rule-based NER application requires patterns which describe the internal structure of names and context-sensitive rules which give clues for classificiation. Looking up name lists, identification of aliases, and handling the ambiguity of names are all necessary for identifying internal and external evidence.

The structure of this section is as follows. In Subsection 5.2.1, we give a description of a rule-based NER system we constructed for identifying NEs in the text of Magyar Nagylexikon (MNL), a Hungarian encyclopedia. Subsections 5.2.2 and 5.2.3 enumerate regular expression patterns which can serve as either internal and external evidence for NER. Subsection 5.2.4 concludes the section with a summary of rule-based systems, pointing out their disadvantages.

5.2.1 A Rule-based System for Recognizing Named Entities in Hungarian Encyclopedic Texts

The linguistic workflow of the MNL project aimed at extracting the important pieces of information from the text and structure of the entire encyclopedia, and assign them to ontology classes, thus providing a knowledge representation which could serve as the basis of several web applications. Since important pieces of information are mostly NEs, the main subtask of the project was NER. We extracted person, location, and organization names, titles of artworks, and temporal expressions[1].

For implementing the steps of the linguistic workflow, we used GATE[2], General Architecture for Text Engineering [Cunningham et al., 2011], which is an open source software capable of solving text processing problems. GATE provides built-in tools, such as tokenizers, sentence splitters,

[1]The project ran in 2003–2004, within the boundaries of the Magyar Nagylexikon Kiadó Zrt., the company responsible for editing and publishing the Hungarian encyclopedia, Magyar Nagylexikon. Two consequences follow: first, language processing resources were used as they were available then. Second, results remained unpublished because they were treated confidentially.

[2]We used version 2.1, the most current version in 2003.

and other higher level processing resources, primarily for English. To adapt it to Hungarian, we had to create several language and processing resources for Hungarian. GATE has another built-in functionality, which is language-independent and very useful for rule-based NER: JAPE is a Java Annotation Patterns Engine, which provides finite state transduction over annotations based on regular expressions.

To achieve our goal of annotating NEs in text, several pre-processing steps had to be taken. The complete workflow of NER is as follows:

Tokenization. Tokenization of the entire text of the encyclopedia was performed by GATE's tokenizer module.

Word lemmatization. Language units annotated as words, thus non-NEs, were lemmatized, and full morphological annotation was assigned to them. This was performed by `Jspell`, a Java reimplementation of `hunspell` [Németh et al., 2004].

NE lemmatization. Since list lookup works on literally equal matches, forms of NEs occurring in the text had to be stripped of their suffixes. However, `hunspell` works with a limited lexicon and contains only frequent names and their lemmatization rules. In addition, lemmatization rules for NEs are different from those for common nouns (cf. Subsection 2.3.2). We therefore adopted a new strategy. The lexicon file of `hunspell` was replaced with name lists generated from the inherent XML tagset of the encyclopedia[3]. Affixing rules which can operate on NEs were selected from the original affix file of `hunspell`, separately for each name type. In Hungarian, suffixation of foreign words and names works according to the constraints of vowel harmony, so allomorphs are chosen based on the phonological form of the name in question. For this reason, a new component was added to the NE lemmatizer, which contains phonological and transcription rules for 20 languages.

Gazetteer list compilation. Even though 'gazetteer' originally means geographical directory, in the context of NER the phrase is simply used to indicate a list of names. Gazetteers for each type of NE were generated from the inherent XML tagset of the encyclopedia. Since the text had been tagged manually by human editors, and the original goal of tagging was only to prepare articles for printing, the tags had

[3]The encyclopedia was written and edited in an XML-based editorial system, which contained tags indicating several types of content, e.g. regnal name, place of birth and death, structured according to the rules of a pre-defined DTD.

many errors. For this reason, we checked and cleaned the lists manually. In addition, NEs had to be lemmatized to get exact matches. This was performed by the NE lemmatizer, and suffixed NE forms were changed to their lemmatized form. The gazetteer lists also contained abbreviated forms of NEs.

Transducing. First, GATE's gazetteer module annotates elements of NEs: if a language unit occurring in the text matches exactly one unit in any of the gazetteer lists, it gets annotated with the appropriate tag, e.g. *person first name*, *person full name*, *month*, or *day*. Second, after investigating the text of encyclopedia, we manually created regular expression patterns operating on these annotations. Finally, we used JAPE to build finite state transducers from the patterns and annotate NEs in the text.

Sentence splitting. Before splitting the text into sentences, abbreviations covered by gazetteer items were omitted, so as not to be considered sentence ending elements.

This order of steps differs slightly from that currently applied by statistical NER systems, for which sentence splitting is usually the second preprocessing step after tokenization, since they operate on sentences and need sentence boundaries to recognize NEs. In contrast, our rule-based system operates on patterns of text, so sentence boundaries are not as important as in the case of supervised systems. Another difference is that we lemmatize NEs and non-NEs separately. For this purpose, the system has to distinguish NEs from non-NEs by the time of lemmatization, which requires a kind of combination of POS tagging and NER. As far as we know, there has only been one attempt to resolve POS tagging and NER in a parallel way. Móra and Vincze [2012] emphasize that by exploiting the differences in affixation of proper names and common nouns in Hungarian, joining the two steps may accelerate the identification of NEs.

The performance of our rule-based system was not measured in any of the standard ways, for several reasons. The first Hungarian NE tagged gold standard corpus, the Szeged NER corpus [Szarvas et al., 2006a], which could have been used to evaluate our system, was not created until a few years later, and at any rate, no full conclusions could have been drawn from it, since encyclopedic text is very different from newswire. For financial reasons, the project was cut before achieving success with the linguistic workflow and obtaining full results, but manual checking of the output showed that our system was able to identify and classify a good proportion of NEs in the text of the Hungarian encyclopedia.

5.2.2 Internal Evidence

In this subsection, we enumerate name patterns which provide information about the internal structure of NE types. The types of interest are: person, organization, and location names, and temporal expressions. Indeed, several more types can be handled by similar rules, but our goal here is to demonstrate the kinds of rules that are generally used, not to provide an exhaustive list. We give examples only for Hungarian NEs and patterns matching them.

For representing rules in this and the following subsection, we use standard regular expression notation (for detailed description, see e.g. Jurafsky and Martin [2000]). Expressions referring to gazetteer lists are italicized as well as the *capitalized_word* term, which refers to the regular expression [A-Z][^]+. We did not create patterns for abbrevations, since it is presupposed that gazetteer lists include them.

When constructing the patterns, we assumed that they are used after the lookup in gazetteer lists has taken place and that they can thus operate on annotation provided by the gazetteers. Rules follow each other in order of precedence, so lower level patterns can be parts of higher level ones. For example, PERSON_NAME in Example 5.4 can be built based on the pattern in Example 5.1. Higher level patterns are marked by small capitals.

Person names

In the Western naming systems, a person name typically consists of a family name and a given name. Indeed, one can have more family names, e.g. *Csokonai Vitéz Mihály*, as well as more given names, e.g. *Esterházy Pál Antal*. In Hungarian, the family name comes first, with the given name behind, a fact which can be formulated by the rule:

(5.1) *family_name*+ *given_name*+ → PERSON_NAME

However, family names, particularly those of famous people, can also stand by themselves, e.g. *Csontváry*. Following the first mention of the full name in a document, the given name and the family name can both stand alone, e.g. *Ganz* and *Ábrahám* after mentioning *Ganz Ábrahám*. Thus, we need more patterns which allow names to stand alone:

(5.2) *given_name* → PERSON_NAME

(5.3) *family_name* → PERSON_NAME

Name affixes, for example generational titles, can be used with a person name and are considered part of a name. In Hungarian, generational titles precede the person name, e.g. *ifj. Bethlen István*. For handling these cases, we need to define one further pattern:

(5.4) *generational_prefix* PERSON_NAME → PERSON_NAME

In ordinary newswire text, names of emperors and popes are not mentioned often, but they are frequent in an encyclopedia. Such names consist of a regnal (or papal) name and a Roman numeral. In Hungarian, the number comes first, followed by a dot, e.g. *XIII. Leó*.

(5.5) [IVX]+\. *regnal_name* → PERSON_NAME

Since naming traditions, spelling rules and other properties of names vary across languages, patterns must be constructed individually for each language that the system is expected to work on. For example, in English, the order of name elements is the inverse of that in Hungarian, since given name precedes family name, generational titles are suffixes, not prefixes, and the numbers in the names of emperors are behind the regnal name, e.g. *John Smith*, *H. C. Mansfield Jr.*, and *Leo XIII*. German, Dutch and Italian names frequently include a family name affix, which can be written in two (or more) words, e.g. *Hahn von Rottenstern*, *Henry van der Velde*, but can also be directly attached to the name, e.g. *Niccolo d'Antonio d'Apulia*. Handling such cases also requires construction of more patterns.

Furthermore, there are naming systems differing from the Western tradition where we cannot divide person names into any of the parts described above, e.g. *Abdalláh ibn Abdal-Muttalib* and *Visvanath Pratap Szingh*. And indeed, several person name types exist throughout the world which cannot be recognized by means of similar patterns.

Organization names

Names of companies can be complex phrases, consisting of several words. Identifying organization names such as *Thinking Machines* or *Next* is quite difficult, sometimes even impossible without considering external evidence. However, there are organization names which share a common feature, such as typical suffixes like *Kft.* or *Zrt.*, the Hungarian equivalents of *LLC* and *Ltd.* To handle such cases, we must create lists of typical organization name suffixes and define an appropriate pattern:

(5.6) *capitalized_word+ organization_suffix*

Location names

Location names can also contain typical affixes which can help identify several location types. For example, names of public places have such suffixes as *utca* ('street') and *tér* ('square').

(5.7) *capitalized_word+ street_suffix*

If a capitalized word is preceded by an orientation prefix, it is likely to be a geographical name, e.g. *Észak-Magyarország* ('North Hungary'). In Hungarian, orientation prefixes are always attached to the names by a hyphen.

(5.8) *orientation_prefix-capitalized_word+*

Similarly, if a capitalized word is followed by a common noun which can be found in a list of geographical objects, and is attached by a hyphen, it is surely a place name, e.g. *Margit-híd* ('Margaret Bridge').

(5.9) *capitalized_word+-geographical_object*

Temporal expressions

Regular expressions are very useful for identifying dates (e.g. *1979*) and time expressions (e.g. *7:48*). Here, gazetteer lookup is not as important as in the previous cases, but short lists with the names of months, days, and seasons can be applied (e.g. *december 1.* ('1st December')). Use of time expressions highly depends on text genre. For example, dates in newswire texts usually refer to a limited time interval (e.g. *2000-es évek* ('2000s')), and do not vary as greatly as in encyclopedic texts (e.g. *Kr. e. 3. század* ('3rd century BC')). However, recognizing time expressions in any kind of text by means of regular expressions usually results in better performance than in the case of names. A few example patterns follow:

(5.10) [1-2][0-9][0-9][0-9]

(5.11) [12]?[0-9]:[0-5][0-9] ([PA]M)?

(5.12) *month* [123]?[0-9]\.

(5.13) Kr\. (e|sz)\. [1-9][0-9]*\. (év)?század

(5.14) ([1-2][0-9])?[0-9][0-9]-[eaöo]s év(ek)?

5.2.3 External Evidence

Internal evidence is usually insufficient for recognizing and classifying names. Knowledge about how certain types of names appear in context is also required. In this subsection, we give examples of patterns that serve as external evidence for NE types. In the case of external evidence, we do not know the type or the internal structure of the capitalized word, since we want to classify it by means of contextual clues. For this reason, the term *capitalized_word* is used in patterns instead of higher level constructions, such as PERSON_NAME.

Person names

Several name affixes can be used with person names, which provide information about the person, indicate that the individual holds a position, educational degree, accreditation, office, or honor. These are usually not considered parts of a name, thus they are examples of external evidence. In Hungarian, these affixes stand before the name, e.g. *gr. Esterházy Károly* ('Count Károly Esterházy'), *Madame Sabatier*. So if a capitalized word is preceded by such a personal prefix, it is likely to be a person name:

(5.15) *personal_prefix capitalized_word*

In news, it is typical to give a person's age right after the name in parentheses, e.g. *K. József (42) bevallotta bűnösségét* ('József K., 42, declared himself to be guilty'). In these cases, the family name is usually abbreviated, so the following rule can be formulated:

(5.16) [A-Z]\. *capitalized_word* \([1-9][0-9]?\)

Apposition of person names is also a typical pattern in news. The definitive element is usually a noun phrase, containing a definite article, an organization name and a common noun signing the role, e.g. *Kis, a Gittegylet elnöke* ('Kis, the chairman of Gittegylet'). This pattern, indeed, can also be applied to the recognition of organization names.

(5.17) *capitalized_word*, az? *capitalized_word role*

The list of patterns could be continued almost endlessly. For several text genres, a large number of similar patterns can be defined, but this requires proper investigation of corpora.

Organization names

It is quite frequent to find after the full organization name its abbreviation in parentheses, e.g. *adta hírül a Magyar Távirati Iroda (MTI)* ('Magyar Távirati Iroda (MTI) reported'). Thus, finding acronyms in parentheses after capitalized word(s) indicates that both of them are organization names. The likelihood grows further if one of them is found to be an organization name in a previous gazetteer lookup.

(5.18) az? *capitalized_word*+ \([A-Z]+\)

Location names

After examining several types of text, we found that location names match patterns that are highly genre-dependent. For example, in encyclopedic text, definition sentences usually contain the place of birth and death, and these definition sentences follow a similar pattern in each article, e.g. *Gaetano Donizetti (Bergamo, 1797. november 29. – Bergamo, 1848. április 8.)* ('Gaetano Donizetti (Bergamo, 29 November 1797 – Bergamo, 8 April 1848)'). In the following pattern we presuppose that dates have been recognized previously. This pattern can also be used for recognizing person names.

(5.19) *capitalized_word*+ \(*capitalized_word*+, *date* – *capitalized_word*+, *date*\)

Temporal expressions

To consider elements in the context of a temporal expression as external evidence, we have to distinguish between absolute and relative temporal expressions. If our goal is to recognize an absolute temporal expression, language units standing close to it in the text and making it relative, e.g. *1991 végén* ('the end of 1991') can be considered external evidence. Postpositions typically occurring with temporal expressions can also aid recognition, e.g. *dec. 2. előtt* ('before 2 Dec').

(5.20) [1-2][0-9]+ (vége|eleje|közepe)

(5.21) hónap [1-3]?[0-9]\. (előtt|után)

The list of expressions and postpositions occurring typically in the context of temporal expressions is short, so they can be simply enumerated with OR operators. As can be seen from the example patterns, however, contextual clues are not necessary for recognizing temporal expressions, since their internal structure is transparent enough.

5.2.4 Summary

We presented a rule-based system which allows us to recognize and classify NEs in Hungarian encyclopedic text. The system was developed only for the MNL, and highly depends on its strict format and the lists extracted from its inherent tagset. It is not portable to any other text, thus our system cannot be used for other purposes. This is one of the main shortcomings of rule-based systems.

After investigating rule-based systems and studying internal and external evidence for particular NE types, we can assert that rule-based NER systems have advantages as well as shortcomings. Certain types, such as temporal expressions have such transparent internal structure that they can be easily recognized by means of regular expressions. This is confirmed by the fact that supervised systems also use Boolean-valued features based on regular expressions, i.e. if a language unit matches such a pattern, then it gets assigned a feature such as `isDate=True`.

Combining internal and external evidence results in higher performance. At the time of writing, we have knowledge of only one rule-based NER system for Hungarian, which also makes use of both internal and external evidence. This system has been reported to achieve 82.13% overall F-measure in recognizing person, location and organization names on a 20,000 tokens sub-corpus of Szeged NER corpus [Gábor et al., 2003]. Table 5.1 shows results for each NE type. High precision and low recall values reveal the advantages and disadvantages of rule-based systems. If we define strict rules, we will recognize a low number of NEs, but will achieve high precision. If our patterns are more permissive, we will find more NEs, at the expense of precision loss. It is impossible to cover all patterns in which NEs can be found without covering such patterns which do not contain NEs, or contain a different type of NE.

	gold #	**TP #**	**FP #**	**FN #**	**P (%)**	**R (%)**	**F (%)**
ORG	1461	1078	373	383	74.3	73.8	74.0
LOC	121	86	4	35	95.6	71.1	81.5
PER	98	85	4	13	95.5	86.7	90.9

Table 5.1: Results of a rule-based NER system for Hungarian (TP=true positive, FP=false positive, FN=false negative, P=precision, R=recall, F=F-measure).

As can be seen from the example patterns above, rules often operate on

annotation supplied by gazetteer lookup. In order to find more NEs and increase recall figures, the best way is to increase the size of gazetteers. Thus, rule-based systems highly depend on the size of gazetteers. (See Subsection 6.5.1 for more details.) NER also requires the identification of aliases, or shortened variations of full proper names. Thus, gazetteer lists have to contain abbreviations and acronyms, or new patterns have to be defined to handle them.

Another way to increase recall is writing rules for further patterns and thus cover a greater number of NEs. However, the number of rules is quite large even in a strict system. If there are lower and higher level rules which can be embedded into each other and must be ordered, then it will be quite difficult to incorporate new rules into the system. It is hard to keep track of all rules and a single human error may cause the system to malfunction.

Rule-based systems are highly language-dependent, since patterns require exact text matches and can vary from language to language. Rules must be revised for each new language, just as new rules are necessary for performing NER on new domains. As illustrated above, there are some patterns which are typical of encyclopedic texts only, while others of newswire only. Thus, rule-based systems are also highly domain-specific.

A major problem with NER is the ambiguity of names. Person, location, and organization names can be composed of the same words. For example, the word 'Jordan' can be a first name, a last name, the name of a river, a country name, or part of an organization name. Different types of names can appear in text in similar ways. To handle this phenomenon, a good solution is a rule competition phase that allows selecting the most probable interpretation for a name (e.g. Krupka and Hausman [1998]), resulting in a not clearly rule-based, but rather a hybrid system combining rules and statistics.

5.3 Statistical Named Entity Recognition

In Section 5.2, rule-based NER systems were described. We mentioned that empirical methods returned into NLP in the 1990s, and they have since become the most widely applied techniques. According to Armstrong-Warwick [1993], empirical methods offer potential solutions to several problems in NLP such as

- acquisition: automatically identifying and coding all necessary knowledge;

- coverage: accounting for all phenomena in a given domain or application;
- robustness: accommodating real data that contains noise and aspects not accounted for by the underlying model;
- extensibility: easily extending or porting a system to a new set of data or a new domain.

Robustness and extensibility are still not properly solved, as illustrated in Section 4.1 and in Subsection 4.2.2, respectively. Despite the fact that gold standard corpora used for training and testing are usually cleaned, empirical methods still offer more robustness than rule-based systems do. And indeed, they can be ported to new datasets or new domains, at the expense of some performance loss, but without the need to rewrite the whole system.

Empirical methods used in NLP can be categorized along several dimensions. One of them concerns the style of the algorithm itself.

Most of the recent work in empirical NLP has involved statistical training techniques for *probabilistic models* such as probabilistic grammars and HMMs. These methods attach probabilities to the productions of a formal grammar or the transitions of a finite-state machine, estimating these probabilities based on training data. Test examples are then analysed by derivation from the learned grammar that generates the given string or finding the most probable path through the automaton built. Other empirical methods use other statistics such as the frequency of n-grams appearing in the language data.

However, not all empirical methods use probabilistic models. *Symbolic methods* represent learned knowledge in the form of interpretable decision trees, or logical rules. These symbolic machine learning systems stand closer to rule-based systems in the sense that the target, in both cases, is inducing rules from data. The difference is whether one creates a rule system by hand or by means of a machine learning algorithm. The latter offers the promise of automating the acquisition of knowledge from corpora. Acquired knowledge is represented in a form that is easily interpreted by human developers and is similar to representations used in manually developed systems. Such interpretable knowledge potentially allows for greater scientific insight into linguistic phenomena, improvement of learned knowledge through human editing, and easier integration with manually developed systems [Mooney, 2004].

Another dimension along which empirical methods can be categorized is the type of data required. Learning techniques may be supervised, semi-

supervised and unsupervised. Supervised systems require large amounts of previously annotated data. As illustrated in Chapter 4, manually annotating corpora with linguistic information is a time-consuming, highly skilled and delicate job. Therefore, reducing annotation cost is of key importance. Besides the automated creation of corpora, another approach is to use semi-supervised or unsupervised methods, which do not require large amounts of labelled data.

The underlying question in the case of *unsupervised learning* is what we can learn from raw text. Unsupervised algorithms are not supplied with correct labels for classification, instead they are given only raw text that has been pre-processed only minimally, i.e. split into tokens and sentences. NER can be seen as a clustering problem, where NE classes are based on similarity of context. Words with similar grammatical behavior will be assigned to the same cluster. Depending on the task, one can pick an arbitrary number of clusters, or let the system find the appropriate number of clusters. If the goal is to test how well such methods can recover commonly used NE types, then one should use the same number of clusters as there are NE tags. However, unsupervised methods can also be used for finding new types of NEs. In this case, one can use a much bigger number of clusters to capture correlations that are not captured by classic NE tags.

Since nothing is said about the identity of each cluster, and the model has no preference in assigning cluster 1 to person names and cluster 2 to location names, finding the metrics for evaluation is not straightforward. One of the applied metrics is called 1-Many [LXMLS, 2011], which maps each cluster to the majority tag that it contains. For example, Elsner et al. [2009] use it when mapping their three induced labels (`PER`, `ORG`, `LOC`) to their corresponding gold labels in the MUC-7 test dataset, and then count the overlap. They report 86% accuracy, which is said to be the best score for a fully unsupervised model.

Besides clustering, there are further unsupervised methods used in NER. These rely on lexical resources such as WordNet, on lexical patterns, and on statistics computed using a large unannotated corpus. Cucchiarelli and Velardi [2001] present an unsupervised method using WordNet for context generalization. They suggest that the use of such complementary methods can increase the robustness of any supervised or rules-based NE tagger. They extract syntactic and semantic contextual knowledge from unannotated data, using more fine-grained evidence than supervised systems do.

Unsupervised learning is an increasingly active field of NER research. Besides the already mentioned pragmatic reason, namely that annotated

corpora are scarce resources, several other motivations have pushed research in this direction. From both a linguistic and cognitive point of view, unsupervised learning is useful as a tool to study language acquisition. From a machine learning point of view, unsupervised learning is fertile ground for testing new methods, where significant improvements can still be made.

The distinction between supervised and unsupervised systems is not always clear. In some systems that are apparently unsupervised, one could argue that the human labour of generating labelled training data has merely been shifted to embedding linguistic knowledge and heuristics in the system. There are some systems which cannot be said to be supervised since they do not use manually annotated data, but they do start with manually constructed example lists of NEs, often called learning seeds. This approach is usually called *semi-supervised learning*. For example, Nadeau et al. [2006] present a NER system which requires no human intervention and can handle more than just the three classic NE types. They claim the system is unsupervised, even though it uses a seed of four NEs per list, which bootstraps the learning process. Similarly, Collins and Singer [1999] show that the use of unlabelled data can reduce the requirements for supervision to just seven simple seed rules. This approach gains leverage from natural redundancy in the data, since for many NE instances both the spelling of the name and the context in which it appears are sufficient to determine its type, they argue.

5.3.1 Supervised Named Entity Recognition

Although building semi-supervised and unsupervised systems is an emerging field of NER, the currently predominant technique is supervised learning. In this subsection, we first give an overview of methods used for supervised NER. Systems based on different learning methods are all variants of the supervised approach that typically involve a system that reads an annotated corpus, extracts features, builds model from them, then assigns pre-defined NE labels to tokens or phrases of a previously unseen text. The remaining part of this subsection describes this workflow in detail.

Methods used for supervised learning

Methods used for supervised learning are based on the assumption that datapoints (in the case of NER, tokens) are independent elements of the text. Tokens are identically distributed, and this fact makes it possible to

use the model trained on the training dataset for tagging new, previously unseen datasets. The assumption behind this is that new data has the same probability distribution as the training data.

The standard way to approach the problem of NER is as a *sequence labelling* task, which correspond to a chain structure, for example, the words in a sentence. These kinds of supervised methods are based on the scenario of structured prediction, where inputs are assumed to have temporal or spatial dependencies. HMMs, maximum entropy Markov models (MEMMs) and Conditional Random Fields (CRFs) are sequence classifiers. A sequence classifier is a model whose job is to assign labels to each unit in a sequence.

The HMM is one of the most common sequence probabilistic models and has been applied to a wide variety of NLP tasks. In NER, an underlying learning algorithm emits probability distribution on tags to a token sequence. These tag emission probabilities show the probability of tags given a feature or a set of features. For example, in a highly simplified system where only one feature is used, $P(ORG \mid iscap)$ is the probability of getting `ORG` label when a token is capitalized. In real systems, however, a vector of a large number of features is assigned to each tag. Transition probabilities of a first-order HMM are probabilities of transitions between particular pairs of tags. For instance, $P(LOC \mid ORG)$ is the probability that an organization name is followed by a location name in a sentence. Transition probabilities can be counted in an annotated corpus. To calculate the most probable tag sequence for a whole sentence, several decoding algorithms, e.g. the Viterbi algorithm can be applied. HMMs for NER are used by e.g. Bikel et al. [1997]. The `hunner` system [Varga and Simon, 2007] also uses HMM and Viterbi, as described in Subsection 5.3.2. For more details on HMMs and the Viterbi algorithm, see e.g. Jurafsky and Martin [2000].

Other sequential models are MEMMs and CRFs. Both can be seen as extensions of the maximum entropy model to structured prediction problems. MEMM is an earlier, less successful attempt to perform such an extension [McCallum et al., 2000]. In this model, each node or edge of the Markov model is a locally normalized maximum entropy model. The shortcoming of MEMMs is the so-called label bias problem. MEMMs model the distribution of a label based on current observations and the previous label. In the training phase, when predicting the next label, MEMM uses gold standard labels, while in the test phase, there are no gold standard labels, so the prediction is based on the previous label emitted by the MEMM, which is not necessarily right [Farkas, 2007]. On the contrary, CRFs are globally normalized models, thus they compute the probability

of a whole sentence, not local probabilities [Lafferty et al., 2001]. CRFs are successfully used for NER, they are reported to achieve the highest scores (e.g. McCallum and Li [2003]). In spite of this fact, we do not use CRFs, since training time is an order of magnitude higher than that of maximum entropy models, due to the large number of features used.

On the other hand, maximum entropy by itself is not a sequence classifier; it is used to assign a label to a single datapoint, in the case of NER, to a token. In that sense, maximum entropy models are *token-based*. For more details on maximum entropy, see Subsection 5.3.2. Another kind of token-based methods is the Naive Bayes classifier, which is used for NER, for example, by Zhang et al. [2004], for finding the most typical NEs among all NEs in a document, and by Tanabe et al. [2005], for predicting gene and protein names in documents. Since our results do not rely on Naive Bayes models, we direct the reader to Jurafsky and Martin [2000] for a more complete discussion.

Besides sequential versus token-based categorization, supervised methods may also be broken down along another dimension, namely what they intend to model. From this point of view, there are generative and discriminative methods. *Generative* methods attempt to model the joint probability $P(X, Y)$ that generates the data, where X is the observation set and Y is the label set which the system emits. Instead of maximizing joint probability, as generative approaches do, *discriminative* methods maximize directly the conditional probability $P(Y \mid X)$. The rationale behind this is that one does not need to waste effort on modelling the distribution of input data, if all we want is an accurate estimate of $P(Y \mid X)$, which is what matters for predicition. Table 5.2 sums up the methods mentioned above [LXMLS, 2011].

Token-based	**Sequential**
Generative	
Naive Bayes	Hidden Markov Model
Discriminative	
Maximum Entropy	Conditional Random Fields Maximum Entropy Markov Models

Table 5.2: Summary of methods generally used for NER.

The workflow of supervised learning

Independent of the method used for recognizing and classifying NEs in text, supervised learning approaches all involve the same elementary steps. As described in detail in Chapter 4, supervised systems need a large amount of data previously enriched with linguistic annotation. The algorithm learns its parameters from a *corpus*, and its evaluation also involves comparing its output to a gold standard annotation.

When trying to compute the probability of a sequence, e.g. one containing NEs, evaluation must not be biased by including any sentence from the training data in the dataset used for evaluation. For this reason, supervised learning always starts by *dividing the data* into a training and a test set.

In some cases, we need more than one dataset for evaluation. In the case of shared tasks, for example, organizers first provide a training and a development set, the former for training our model and the latter for testing our system during the development process, for measuring the discriminative power of features, and for experimenting in general. Researchers then submit the system which is proved to be the best on the development set. The test set is usually provided just before submission deadline and is used to measure the performance of systems. It is not allowed to run the system on the test set more than once. This is not simply a case of fair play; using the test set as a development set and tuning the system for achieving better scores on it will not illustrate the real performance of the system. This phenomenon is known as *overfitting*. Overfitting occurs when the model adapts to random noise present in the training data rather than learning to generalize from its parameters.

In the absence of a development set, the method of n-fold *cross-validation* is often used to train and evaluate on the same corpus. One round of cross-validation involves partitioning the whole dataset into complementary subsets, training on one subset, and testing on the other one. Multiple rounds of cross-validation are performed using different partitions, and the evaluation results are averaged over the rounds. 'n-fold' means that the whole dataset is divided into n parts, and usually $n-1$ nth of the data is used for training, and 1 nth is used for testing. This is repeated n times, with a different partition used for evaluation in each run. Thus, cross-validation is a way to predict the fit of a model to a hypothetical evaluation set when an explicit evaluation set is not available.

When building a supervised machine learning system, a major step is *feature extraction*: collecting information from the data that can be relevant to the task. In the case of NER, such pieces of information can be simple

orthographic properties, morphological information, as well as gazetteer list inclusion of the inspected datapoint and other datapoints in its neighbourhood. At the end of the featurizing step a feature vector containing all relevant information is assigned to every datapoint.

The next step is *model building*, where the task of the algorithm is to find regularities in this large amount of information. Since the training data contains datapoints with gold standard labels, the algorithm builds a model based on feature–label pairings.

In order to *tag* the test set with the appropriate labels, the model must have access to feature vectors of datapoints in the test set. Therefore, features used for training must be extracted from the test set. Given the feature vectors, the model can predict labels for the new datapoints.

As mentioned above, *evaluation* of NER systems is performed by comparing the system's output to gold standard labels. The standard evaluation metrics are based on the following terms:

true positive: the system outputs the correct NE label;

true negative: the system correctly outputs ○, i.e. it recognizes that the inspected datapoint is not a NE;

false positive: the system outputs unexpected result, i.e. it identifies a non-NE as a NE;

false negative: the system does not recognize a NE.

Based on these values, precision and recall are calculated over all NE slots. They are defined as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The measure combining precision and recall is their weighted harmonic mean, the F_β-measure:

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}$$

In NER, β is usually equal to 1, thus recall and precision are evenly weighted. This is called F_1-measure. Depending on the application, F-measure can be used with higher or lower β, where the former weights

recall higher than precision, while the latter puts more emphasis on precision.

Another type of error in NER is when the system does not find the correct boundaries of a NE. Scoring techniques differ in handling this kind of error. In MUC evaluations, a system is scored on two axes: its ability to find the correct type and its ability to find the exact boundaries of NEs in the text. The final F-measure is then calculated on both axes, thus type and boundary scores are summed up. This scoring system gives partial credit to errors occurring on one axis only, while missing both type and boundary results in a double loss. The CoNLL scoring protocol, on the other hand, follows the so-called exact match evaluation. In this protocol, a prediction is judged to be correct only if it has the same type and the same start and end positions as the gold standard datapoint. The CoNLL F-measure is more strict, thus these values are lower than those in MUC evaluation. This is why systems competing in MUC shared tasks apparently achieve higher scores (93.39%) than those in CoNLL (88.76%).

In shared task evaluation, a *baseline* system is usually presented to determine the minimal expectations for performance. Several baseline counting methods are in use, but the most widely used one is when every NE has the same label, namely the most frequent one among all NEs. This method is used e.g. by Elsner et al. [2009] and by Markert and Nissim [2007b] in the metonymy resolution shared task.

5.3.2 Hungarian Named Entity Recognition with a Maximum Entropy Approach

In this subsection, we describe `hunner`, a language-independent NER system applied to Hungarian. Since its first version [Varga and Simon, 2006, 2007], the system has evolved in various ways. It was reimplemented and thereby generalized to other NLP tasks besides NER. In a next step, the underlying optimization system that used L-BFGS algorithm [Zhu et al., 1997] was replaced by the `Liblinear` classifier [Fan et al., 2008]. This version was renamed, and it is now available as `huntag`[4]. In the subsequent description, we present the original system[5] with notes on modifications made during the reimplementation. We always refer to the system as `hunner` when writing about its application to the NER task.

For major languages, hundreds of papers were published on NER algorithms, not many of which have language-dependent components. For

[4] `https://github.com/recski/HunTag/`

[5] The system description is based on our article [Varga and Simon, 2007].

Hungarian, we are aware only of one quantitative study of a NER system based on machine learning methods. Szarvas et al. [2006b] published results on their NER system based on C4.5 decision trees with boosting. Their system achieved state-of-the-art performance for English, and reached 94.77% CoNLL F-measure for Hungarian.

Architecture

Our system roughly follows the architecture described by Borthwick [1999] and Chieu and Ng [2003], incorporating some ideas introduced by Klein et al. [2003]. We use the *maximum entropy method*, which has been already successfully used in several NLP tasks. As shown in Subsection 5.3.1, the maximum entropy method is discriminative, and it focuses on finding a separating hyperplane to discriminate among classes. It provides a token-based classification, thus feature vectors are assigned to tokens, not sequences. However, information on the immediate context of a token are not lost with this approach, since they can be added as contextual features to the inspected token's feature vector.

We have chosen Zhang Le's maximum entropy implementation[6] because of its high performance. This implementation uses the *L-BFGS algorithm* [Zhu et al., 1997] for model building. There are two different formalizations of the learning task, both leading to the maximum entropy method. One formalization is based on the concept of entropy. In this case, the basic rationale is that choosing the model with the highest entropy corresponds to making the fewest possible assumptions regarding what was unobserved, trying to make uncertainty of the model as high as possible [LXMLS, 2011]. The L-BFGS algorithm follows this approach for model building. The second formalization treats the question as a linear regression problem. The `Liblinear` library supports this approach. Since the first runs showed that overall F-measures achieved by using `Liblinear` are not significantly better than those achieved by L-BFGS, `Liblinear` requires more memory, and its running time is much longer, here we deal with the former approach.

L-BFGS is an iterative learning algorithm, which starts to converge after approx. 100 *iterations* on our datasets. Our originally published numbers are based on 300 iterations. Later experiments proved that when training on a typical gold standard dataset such as the CoNLL-2003 or the Szeged NER corpus, 200 iterations are generally sufficient. However, when using a much larger dataset such as the Wikipedia corpus, more

[6] `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`

iteration steps are required by the model to reach the highest training accuracy. In the original system, the running time of 300 iterations was quite high (approx. 1 hour), but after the reimplementation, it was reduced to a few minutes. One of the great advantages of maximum entropy modelling is that it is much faster than, for example, its sequential counterpart, the CRF.

We use a very high total number of features. The 200,000 token corpus is represented by 10 million instances of 250,000 different kinds of features. The maximum entropy approach is capable of dealing with such a high number of features without the feature selection phase needed by some other machine learning methods. Nevertheless, the system has a function by which the number of training events, i.e. the feature space can be reduced. This parameter is called *cutoff*. If the cutoff is equal to 1, then every feature occurring at least once is considered in model building. Since the maximum entropy method is capable of handling enormously high number of features, we do not need to set this parameter higher. However, if setting a higher value causes significant decrease in performance, this suggests that the model is overfitted on the training data.

The algorithm builds a model based on feature–label pairs, with each pair being assigned a weight showing how the feature changes the likelihood that a token will get that label. To avoid overfitting, we can add a *Gaussian prior*, which penalizes large weights. This regularization term was not part of the original system, so our results given here are not effected by any penalty.

The NER task in its original form deals with the classification of unknown contiguous token sequences, and it is not immediately obvious how to formulate it as a token classification task. Instead of using classic CoNLL labels (`O, LOC, ORG, PER, MISC`), we chose the following solution: every token must be classified into one of 17 different classes: {`O, B-LOC, I-LOC, E-LOC, 1-LOC, B-ORG, ..., 1-MISC`}, where `B`, `I` and `E` mean the beginning, interior, and end of the NE, respectively, and 1 means that the word is in itself a NE. There are two major advantages of this approach. First, the machine learner can more easily recognize correlations that are specific to the start or the end of NEs. Second, the tagset has implicit built-in consistency requirements: e.g. `B-*` cannot follow `I-*`.

One important characteristic of maximum entropy learning is that during classification it emits a full probability distribution on tags, instead of just the single most likely tag. This gives us the ability to override local decisions if they prove to be inconsistent with each other. First we query the model built by the maximum entropy algorithm for tag emission probabilities for each token. We then define transition probabilities between

tags as follows: transition probabilities for illegal transitions (e.g. `B-ORG B-LOC`) are set to be zero, and every legal transition (e.g. `B-ORG I-ORG`) is set to be equiprobable. Since here we investigate tag bigrams, this construction can be seen as a first-order HMM, on which we can apply a *decoding* algorithm to find the most probable tag sequence. We use the Viterbi algorithm, which maximizes the joint transition-emission probability for a whole sentence. Thus the resulting tag sequence is necessarily well-formed. According to our measurements, this parameterless post-processing step improves the system's F-measure by approx. 0.5% in typical measurement setups.

Table 5.3 shows an example token sequence from the Szeged NER corpus with its gold standard and predicted labels and probabilities emitted by the algorithm. *C1* and *C2* mark predicted classes, while *P1* and *P2* are probabilities assigned to those classes. If we did not use any decoding algorithm, but took into account only the highest local probabilities, we would get the following tag sequence: `O B-ORG I-ORG B-ORG E-ORG O`. When using the Viterbi-algorithm, illegal transitions are not allowed, so two competing consistent hypotheses remain: `O B-ORG I-ORG I-ORG E-ORG O` and `O 1-ORG O B-ORG E-ORG O`. After calculating joint transition-emission probabilities, the first sequence is computed to be the most probable, and it is the right choice.

Token	**Gold**	**C1**	**P1**	**C2**	**P2**
az	O	O	1		
Investicna	B-ORG	B-ORG	0.92	1-ORG	0.08
a	I-ORG	I-ORG	0.96	O	0.04
Rozvojova	I-ORG	B-ORG	0.65	I-ORG	0.35
Banka	E-ORG	E-ORG	1		
(	O	O	1		

Table 5.3: An example token sequence to illustrate how the Viterbi-algorithm works.

During the reimplementation of the system, there has been a change in the way transition probabilities are defined. Instead of manual setup, in the current version of the `hunner` system, transition probabilities are calculated based on the training corpus. The system has a built-in function to build a *language model* using some annotation of the training data. In the case of NER, this annotation is obviously the gold standard NE labelling. Any corpus can serve as a training corpus which has NE labels in `BIE1`

format, but usually the same corpus is used for building language model as for training the observation model. In its current version, the language model is based on label bigrams, but this can be expanded even to trigrams or more-grams. However, as our experiments show, the bigram model suffices for decoding on NE labels. The weight of the language model in the prediction of NE labels can also be configured. If a higher value is given to the language model, the contribution of transition probabilities will be more than that of the probabilities emitted by the maximum entropy algorithm. Based on tests on the CoNLL-2003 development dataset, tagging with a balanced language model weight (`lmw=1`) gave the best overall F-measure.

Feature extraction

Chapter 6 gives a detailed description of several types of features generally used in NER. Here we simply enumerate the features used in the original system.

Most of our features deal with very easily computable surface properties of tokens, but we also use morphological information of tokens and gazetteer lookup features. The complete list of features used by our original system is as follows:

The token's position in the sentence: sentence start, sentence end.

Boolean-valued surface properties of the word form: upper case, all upper case, contains capitalized letter after non-capitalized (camel casing, e.g. *iPod*), is a number, contains a number, contains a dash, contains a period.

String-valued surface properties of the word form: the word form itself, the five-letter prefix, and all consecutive character trigrams of the word form.

String-valued morhological information: POS tags and lemmas provided by the `hundisambig` morphological disambiguator.

Boolean-valued features concerning morphology: is the word form recognized by `hundisambig`? Is the identified lemma capitalized differently than the token itself?

Gazetteer features: if the token is included in one of the gazetteers, it receives a feature containing the name of the list and also information about whether the token is in the beginning, middle or final position

of a multi-word name (e.g. `firstname=start, street=mid`). If the token by itself matches to a one-word name, it receives a `*=lone` feature. To deal with morphology, when determining whether a word matches a multi-word name, the last word was treated differently: matching on a suitably chosen prefix was enough. This corresponds to the way Hungarian multi-word names are inflected.

For example, the *Gyula* token in a sentence starting position gets the following features:

- Boolean-valued surface features: `sentstart=1 sentend=0 iscap=1 allcaps=0 camel=0 isdigit=0 hasnumber=0 hasdash=0 hasperiod=0`
- String-valued surface features: `form=Gyula prefix=Gyula ngr=Gyu ngr=yul ngr=ula`
- Morphological features: `pos=NOUN lemma=Gyula`
- Gazetteer features: `firstname=lone city=lone familyname=lone corp=start`

The system collects these pieces of information for each individual token of a sentence. To incorporate context, we simply add features of neighbouring tokens, recording their relative positions. For example, if a token gets the feature `-2_iscap=1`, it means that the token two positions before is capitalized. The size of the context window for a given feature is a parameter of our system, which can be set for each feature separately. In the original system, we used a context radius of 3 (that is, a seven-token interval) for character n-grams and prefixes, and context radius 5 for the rest of the features.

Gazetteers

We assembled various gazetteers to be incorporated into our system.

- Hungarian and common non-Hungarian last names
- Hungarian and common non-Hungarian first names
- names of Hungarian cities
- country names in Hungarian

- Hungarian street names
- Hungarian organization names
- international organization names
- suffixes for company names
- suffixes for street names
- financial acronyms

In the first six cases, our source was an aggregated version of a Hungarian phone book and a web database. The lists of Hungarian organization names and street names were cleaned of suffixes with automatic methods. Common suffixes were extracted and moved into separate lists. The international organization list was kindly provided to us by György Szarvas and Richárd Farkas.

The gazetteers incorporated into our system were finalized before the inspection of the training and development corpora. During the tuning of the system to the development corpus, we have found serious cases of over- and undergeneralization in the gazetteers. Since correcting these errors did not improve performance significantly, we reverted to the original, uncorrected, automatically collected versions.

There was only one case when analysis of the development corpus lead to the inclusion of a new dictionary: the lexicon of financial acronyms. The development corpus contains several stock market index names (e.g. *DAX, Libor, Nasdaq*), which were sometimes marked as `ORG` instead of `MISC` by the model. To solve this problem, we extracted such stock market terms from a web-based financial knowledge base. We note that using this lexicon did not improve the performance on the test corpus, and even decreased it slightly. The reason for this is that most of these terms occurred only in the development set.

Similarly to the source code of the system, we published the gazetteers under a free document license [7].

Evaluation

We started early development of our system with an ad hoc train–test split of the Szeged NER corpus. But it quickly became apparent that if we intend our results to be comparable to the only existing quantitative study

[7] They are available via the URL `http://krusovice.mokk.bme.hu/~eszter/ner_listak`.

on Hungarian NER, we have to switch to the train–development–test split used by Szarvas et al. [2006b]. They were kind to provide this split, and from this point, we followed the standard methodology: we optimized the parameters of the system guided by the F-measure on the development corpus, and only evaluated on the test corpus once, after the optimization phase was finished.

Table 5.4 shows the results of our original system for Hungarian with the architecture and feature set described above. We reached a CoNLL F-measure of 96.35% on the development corpus and 95.06% on the test corpus. This is a minor improvement on the numbers published by Szarvas et al. (we do not have access to their development results for each NE type).

NE type	**Szarvas devel**	**Szarvas test**	**devel**	**test**
LOC		95.07	92.06	96.36
MISC		85.96	93.58	85.12
ORG		95.84	97.62	96.20
PER		94.67	97.44	94.94
Overall	96.20	94.77	96.35	95.06

Table 5.4: Results of the original `hunner` system on the Szeged NER corpus, compared to Szarvas et al.'s results.

We measured the effect of each major subset of features. The overall F-measure of the system is 95.06% on the test corpus. Removing just the Boolean-valued surface features decreased this score to 92.37%. Removing just the character n-gram features decreased the score to 90.04%. The gazetteer and morphological features had significantly less effect: removing these decreased the score to 94.69% and 94.70%, respectively. Note that these two sets of features are exactly the ones that require external resources. Removing both of them lead to a resourceless system without seriously affecting the score: the resourceless system had an F-measure of 94.73%.

As can be seen from this description, our system is a modularized one, thus the components can be changed if needed, and none of the modules are language-dependent. The only elements which require external resources are the morphological and gazetteer features, and removing them does not cause a significant decrease in the performance for Hungarian. Thus, our system can also be used for recognizing NEs in other languages. Moreover, it can be applied to several other NLP tasks. Its reimplemented

version has been used for English NER [Simon and Nemeskey, 2012] (cf. Subsection 4.3.4), for recognizing metonymic NEs in English [Farkas et al., 2007] (cf. Subsection 3.3.2), and for classification of semantic relations between pairs of nominals [Hendrickx et al., 2010]. In addition, it has been also used for shallow syntactic analysis of Hungarian texts [Recski and Varga, 2010], but we did not contribute to that work.

task	**train**	**test**	**F (%)**
Hu NER	Szeged_wikilists	Szeged	95.48
En NER	CoNLL_wikilists	CoNLL	86.34
En metonymy resolution			
loc-coarse	SemEval-2007	SemEval-2007	85.20
org-coarse	SemEval-2007	SemEval-2007	76.70
En semantic relations	SemEval-2010	SemEval-2010	66.33
Hu chunking	Szeged Treebank	Szeged Treebank	89.87

Table 5.5: Best overall F-measures achieved by our system on several tasks.

Table 5.5 summarizes the best overall F-measures achieved by our system on several tasks. The results for NER and metonymy resolution are repeated from Chapters 4 and 3, respectively. The classification of semantic relations between pairs of nominals was a SemEval-2010 shared task, to which our system was submitted. The results published here (66.33% F-measure on the test set provided by the organizers) was the 8th in the competition, so we decided not to write a system description paper. The task description and results of all submitted systems are reported in Hendrickx et al. [2010]. The result is mentioned here only to illustrate the wide variety of NLP tasks for which our system can be used. And last, but not least: the reimplemented `huntag` system is applied for shallow parsing under the name `hunchunk` [Recski and Varga, 2010]. For Hungarian, it was trained on the Szeged Treebank [Csendes et al., 2005].

5.4 Conclusion

The task of NER can be approached in two ways: by manually coding regular expression patterns or by automatically extracting relevant information and using some machine learning technique. Both have advantages as well as disadvantages. Manually developing a rule-based system and

handling a large number of rules is quite difficult, requiring a great deal of domain-specific knowledge engineering. In addition, these systems are brittle and not portable between different domains or tasks. Empirical methods, on the other hand, offer potential solutions to several problems in NLP, e.g. knowledge acquisition by means of automatic learning techniques, coverage by means of large amounts of data, robustness by means of frequency-based algorithms, and extensibility by means of portable systems. Although these problems are still not properly solved, using empirical methods results in higher performance.

The current dominant technique used in the field of NER is supervised learning. Its disadvantage is that it requires large amounts of previously annotated data, so one might say that the human labour of creating rules has only been shifted to that of building corpora. However, there is a quite new, emerging field of NLP and of NER in particular, which involves using unsupervised or semi-supervised techniques. In these cases, human labour has also been shifted to constructing seed examples and/or embedding heuristics in the system. Unsupervised learning is a field where significant improvements can be made in the future. Another future direction is hybridization, combining the strengths of rationalist and empiricist methodologies.

Chapter 6

Feature Engineering

Features are descriptors or characteristic attributes of datapoints in text. In supervised learning, feature vectors are assigned to datapoints, each of which contains one or more Boolean- or string-valued features. Feature vector representation is a kind of abstraction over text. The task of the algorithm is then to find regularities in this large amount of information that are relevant to the classification task; in this case, to NER.

NER is a typical sequence labelling task, i.e. the model has to assign NE labels to sequences, e.g. sentences in text. Several machine learning algorithms, however, such as maximum entropy modelling, realize it as a token-based classification task. Contextual information is not lost, since the features of neighbouring tokens can be added to the inspected token's feature vector.

In this chapter, we present the features most often used for NER in the token-based classification scenario. We categorize features along the dimension of what kind of properties they provide: surface properties, digit patterns, morphological or syntactic information, or gazetteer list membership. We also study the effect of gazetteer list size on the performance of NER systems.

Defining features for a supervised system is manual work, similar to coding patterns for a rule-based system. However, in the case of statistical methodology, it is the data and not the linguist that determines the usefulness of a feature. The human cognition tends to realize only salient phenomena and will regard as important properties some that are then shown to be unimportant by corpus data and conversely, will fail to notice important ones. For this reason, the power of features has to be measured on real data before inclusion into the system. This is called feature engineering.

6.1 Methods

To measure the strength of features, we build NER systems for Hungarian and English in parallel, adding new features one by one. If the inspected feature is useful, we retain it and add the next one to the system. We consider a feature useful, if adding it does not decrease the performance. The rationale behind this decision is that if a feature does not make things worse on the development set, it may be effective on the test set. Figures indicating F-measures achieved by the system after adding a useful feature are italicized in tables.

There are several feature selection methods used in machine learning which select of a subset of relevant features for inclusion in the model. The simplest algorithm is to test each possible subset of features and finding the one which maximizes the F-measure. An exhaustive search of the feature space, however, is generally impractical. Since we work with a large number of features and can rely on results from previous experiments, we decided to choose an incremental method of feature selection. In this scenario, we start with an arbitrary feature subset, then attempt to find a better solution by incremental extension of the subset with new features. After each step, the system's performance is evaluated and compared to the previous results. The feature subset achieving the best overall F-measure is then retained.

Selection of the initial feature of each feature category is based on previous experiments: the strongest feature is added first. In this feature selection scenario, however, there are some features which would not be removed if they were added to the system in a different order. Since these features cause insignificant changes in performance, we are probably not off the track when removing them.

As described in Subsection 5.3.2, the `hunner` system has built-in parameters for setting the number of iterations, the Gaussian penalty, the weight of the language model, and the number of features used for model building (cutoff). We set these parameters so as to keep the system as neutral as possible. Thus, we do not apply Gaussian penalty and set the cutoff to 1, which means that all features are taken into account in model building. The language model weight is set to 1, thus the emission probabilities of the maximum entropy model and the transition probabilities of the HMM have equal weights. All experiments are run with 110 iterations, which has proven to be sufficient even with such a large number of features.

We use the reimplemented `hunner` system described in Subsection 5.3.2. Since it is language-independent, it can be used for Hungarian and

English in parallel. For Hungarian, we use the same train–development–test split of the Szeged NER corpus that was used for the evaluation of the original `hunner` system. For experiments on English, we use the standard train–development–test sets of the CoNLL-2003 corpus. Unless mentioned otherwise, the results should be interpreted as standard F-measures (%) achieved on these development datasets. Finally, we evaluate the best feature combination on the corresponding test sets and measure the strength of each feature subset.

6.2 Surface Properties

A wide range of features are related to the character makeup of tokens, which are very easily computable surface properties. Surface features can be divided into two main classes: string-valued and Boolean-valued surface properties. These features are described in the next two subsections.

6.2.1 String-valued Surface Properties

Several applications use lists extracted from training corpora as part of a language model. Chieu and Ng [2003], for example, derive lists of frequent and rare words, typical words that precede instances of a name class, and function words. These list features have effects similar to those of using the token's word form or its substrings as features (features based on explicit gazetteer lists are discussed in Section 6.5). Klein et al. [2003] present a NER system based on character n-grams of tokens only. These experiments confirm that using the token's word form and its substrings as features is very useful for NER.

We applied similar features: using only the word form of tokens in the training corpus as features in a 3 tokens radius, and nothing else, results in a 90.68% F-measure for Hungarian. The same run for English gives a result of 80.85%. Setting the radius to 2 increases the score to 81.79%. This difference can be attributed to the fact that Hungarian is a highly agglutinative language and the learning algorithm needs more training data to deal with the great variety of word forms. From here on, the default radius values will be fixed at 3 for Hungarian and 2 for English.

Adding character n-grams to the set of features results in a significant increase in performance both for Hungarian and English. Table 6.1 presents the results of experiments with bi-, tri-, and four-grams. Using bigrams gives the best result, we therefore retain it: the next run uses word form and bigram features, to which we add the n characters prefix

of the token as a new feature. This feature was also applied in the original `hunner` system as a simple substitute for lemmatization. Since adding the prefix feature increases the F-measure for both languages, it is kept as a useful feature.

feature	**Hungarian**	**English**
word form	90.68	81.79
+ngram		
2gram	*93.39*	*83.54*
3gram	92.24	83.51
4gram	92.12	83.41
+prefix		
4prefix	93.30	84.57
5prefix	*93.66*	84.74
6prefix	93.21	*84.90*
+suffix		
3suffix	92.85	85.95
4suffix	93.12	*86.31*
5suffix	93.21	85.97
6suffix	93.38	85.86
+longpatt	*94.92*	*87.21*
+shortpatt	*95.10*	*87.41*

Table 6.1: Results of adding string-valued surface features one by one to the system.

Suffixes are also important from the point of view of morphology. Some word endings can aid recognition of several types of NEs. For example, in English, names of nationalities and languages often end in *-ish* and *-ian* (e.g. *English*, *Hungarian*). It is interesting that increasing the number of characters in prefix and suffix features has the inverse effect on the performance for the two languages. While longer prefixes increase performance for English, the increase stops after five characters for Hungarian. Suffixes show the opposite behaviour, i.e. they can get lengthened arbitrarily for Hungarian only. Increasing this parameter above a certain value would eventually be equivalent to the duplication of the word form feature. The linguistic explanation behind this phenomenon may be the high variability of word endings in Hungarian: there are no typical suffix classes stand-

ing out, at least in such sparse data, while four characters are enough for typical English endings to appear. Since the suffix feature decreases the overall F-measure for Hungarian, it is removed from the Hungarian system. For English it is retained, its length fixed at four characters.

Pattern features were introduced by Collins [2002] and later used by others [Cohen and Sarawagi, 2004; Settles, 2004]. Their role is to map tokens onto a smaller set of patterns over character types. We use these features in both a longer and a shorter, condensed version. The long pattern feature maps all upper case characters to 'A', all lower case ones to 'a', and the rest to '_'. For example, the token *MTI-t* ('MTI.ACC') is mapped to 'AAA_a'. In the short version, consecutive character types are not repeated: 'A_a'. Both versions proved useful, so we retain them.

6.2.2 Boolean-valued Surface Properties

There is quite a wide range of surface properties which can be described by Boolean-valued features. They are formalized as regular expressions, thus, if a token matches the pattern, a feature like `iscap=1` will be assigned to it. Boolean-valued surface properties can be divided into further subcategories. First, we add features to the system that concern *casing*. Similar features are used by most NER systems (e.g. Bikel et al. [1999]; Mayfield et al. [2003]). The features are as follows:

`hascap`: the token contains one or more upper case letters, e.g. *EasyJet*;

`allcaps`: contains only upper case characters, e.g. *XP*;

`capperiod`: comprises one upper case letter and a period, e.g. *A.*;

`camel`: is in camel case, e.g. *iPad*;

`3caps`: comprises three upper case characters, e.g. *IBM*;

`iscap`: its initial letter is upper case, e.g. *London*.

Note that some of these features overlap. For example, the token 'iPad' is assigned both the `camel=1` and the `hascap=1` features. This does not cause problems since maximum entropy models are designed to handle feature overlap. While a high degree of overlap is theoretically not harmful, it has practical disadvantages: it may slow down run-time performance, and may require a higher number of iterations [Borthwick, 1999]. At this point in building our system, training accuracy starts to converge to

its maximum after the 42nd iteration, and the whole featurizing–training–testing process takes only 4 minutes altogether. Therefore, we do not need to be concerned with the problem of overlapping features.

feature	Hungarian	English
best so far	95.10	87.41
+hascap	*95.29*	87.14
+allcaps	*95.29*	86.95
+capperiod	*95.30*	87.27
+camel	*95.30*	*87.41*
+3caps	94.92	87.10
+iscap	*95.46*	86.25

Table 6.2: Results of adding Boolean-valued surface features concerning casing one by one to the system.

As can be seen from the figures in Table 6.2, adding these features one by one slightly increases the performance of the Hungarian system. Only the `3caps` feature decreases the F-measure, which we therefore discard. None of these features improves the English system, however. The `camel` feature alone does not decrease the performance, since there are no tokens in camel case in the development set. However, this may be different in the test set, so we keep this feature, while all other features concerning casing are removed from the English system.

The second category of Boolean-valued surface features is that of *digit patterns*. Digits can express a wide range of useful information such as dates, percentages, intervals, or identifiers. Digit pattern features are particularly useful for tasks like the MUC shared task, which aim at recognizing these elements of text. In the case of CoNLL datasets, where dates are of no interest, digit pattern features can predict that the token matching a typical date pattern is probably not a NE. Similar features are widely used in NER, for example by Bikel et al. [1999] and Zhou and Su [2000]. The features are as follows:

`2digit:` the token is a two-digit number, e.g. *22*;

`digitcomma:` comprises digits and a comma between them, e.g. *200,000*;

`4digit:` is a four-digit number, e.g. *1979*;

`digitdash:` comprises digits and a dash, e.g. *2011-12*;

hasdigit: contains a digit anywhere, e.g. *Boeing-767*;

startswithdigit: starts with a digit, e.g. *2000-ben* ('in 2000');

digitslash: comprises digits and a slash between them, e.g. *2011/12*;

isdigit: contains only digits, e.g. *11*;

1digit: is a one-digit number, e.g. *4*;

yeardecade: is a two- or four-digit number followed by an *s*, e.g. *80s*.

feature	Hungarian	English
best so far	95.46	87.41
+2digit	*95.73*	86.99
+digitcomma	*95.91*	87.16
+4digit	*95.91*	86.98
+digitdash	95.64	87.40
+hasdigit	*95.91*	87.30
+startswithdigit	95.37	*87.73*
+digitslash	95.55	87.60
+isdigit	*95.91*	87.45
+1digit	95.73	87.17
+yeardecade	*95.91*	87.65

Table 6.3: Results of adding Boolean-valued surface features concerning digit patterns one by one to the system.

Table 6.3 shows the results. Performance on the Hungarian data reaches the score of 95.91% after adding the `2digit` and `digitcomma` features, after which none of the newly added features increase this value. The `4digit`, `hasdigit`, `isdigit`, and `yeardecade` features do not change the overall F-measure, since there are no tokens in the training data that match these patterns. We remove those features that decrease the F-measure and keep those which do not have a negative effect. Quite counterintuitively, only one digit feature, the `startswithdigit` improves the F-measure on the English data.

The third category of Boolean-valued surface features are concerned with *punctuation*. The features are as follows:

hasdash: the token contains a dash, e.g. *MTI-t* ('MTI.ACC');

hasperiod: contains a period, e.g. *Corp.*;

punct: is a punctuation mark, e.g. *?*.

feature	Hungarian	English
best so far	95.91	87.73
+hasdash	95.29	87.36
+hasperiod	95.27	86.96
+punct	95.02	87.26

Table 6.4: Results of adding Boolean-valued surface features concerning punctuation one by one to the system.

As can be seen from the figures in Table 6.4, defining such punctuation features decreases the performance of the system. Thus, these features are not important ones and are removed from both the Hungarian and the English system.

6.3 Morphological Information

Some simplified morphological information has already been given to the system by prefix and suffix features, but we have the option of adding more sophisticated features concerning morphology. Since Hungarian is a language with a complex morphology, we expect that adding these features will improve performance. Some morphological features are also usually applied for English NER.

Hungarian morphological information (POS tag, lemma, and full analysis) is provided by `ocamorph`, an Ocaml reimplementation of `hunmorph` [Trón et al., 2005a], based on Hungarian affix and lexicon files of `morphdb`, a lexical database and morphological grammar [Trón et al., 2006a]. `Ocamorph` has a built-in guessing functionality, which indicates the unknown words as out-of-vocabulary (OOV) and assigns the highest ranked candidate analysis to them. Selecting the best fitting one of multiple analyses was performed by `hundisambig`, a statistical morphological disambiguator [Halácsy et al., 2005].

The original CoNLL-2003 dataset contains Penn Treebank POS tags [Marcus et al., 1993], but does not contain the lemma of tokens. These were obtained using `ocamorph`, whose output was mapped to the Penn tagset. Lemmas were pasted to the original dataset in a new column and expanded with the 'OOV' code if the word was not found in the English lexicon of `morphdb`.

Morphological features are as follows:

`lemma`: the token's lemma;

`fulltag`: full morphological analysis;

`pos`: only the POS tag without information about inflection;

`tagend`: information about inflection;

`oov`: out-of-vocabulary word;

`tagpattern`: pattern of POS tags in a sentence in 5, 7, and 9 word windows;

`isbetweensamecases`: whether the inspected token's neighbours have the same case marking as the token itself;

`penntags`: an abstraction over Penn tags which groups together similar tags, e.g. tags starting with *VB* and *MD* get the value 'verb';

`plural`: whether the token is in the plural form.

Table 6.5 shows the results of adding morphological features one by one to the system. Adding the lemma as a feature results in a slightly higher F-measure on Hungarian data, but none of the other morphological features increases the performance before `oov`. The information that the word is not included in the morphological analyser's lexicon seems important for both languages. This is not surprising, since such lexicons usually do not contain NEs, except for a few very frequent ones. The `tagpattern` and `isbetweensamecases` features provide information about the broader neighbourhood of the token, thereby carrying pieces of information about sentence structure. In the case of `tagpattern`, we override the default radius (3 for Hungarian, 2 for English), and the feature is tested on both broader and narrower windows. The 7 word window (radius=3) results in the best scores for both languages. Since in English there is no case marking, the `isbetweensamecases` feature is applied only to Hungarian. Similarly, the `penntags` feature can be interpreted only on

the English data, so it is added only to the English system. When recognizing metonymic NEs, the `plural` feature proved to be very useful (see Subsection 3.3.2), so it is also applied here.

feature	**Hungarian**	**English**
best so far	95.91	87.73
+lemma	*95.99*	87.50
+fulltag	95.64	87.43
+pos	95.37	87.02
+tagend	95.02	87.47
+oov	*95.99*	*87.87*
+tagpattern		
2rad	96.09	87.57
3rad	*96.35*	*87.89*
4rad	95.99	87.87
+isbetweensamecases	*96.79*	-
+penntags	-	*87.90*
+plural	*97.14*	*88.02*

Table 6.5: Results of adding morphological features one by one to the system.

Instead of the guesser functionality of the morphological analyser, one can use more sophisticated methods for lemmatization of Hungarian NEs. For example, Farkas et al. [2008] presents a web-based method for stripping several suffixes of NEs. As can be seen from the results, the morphological features presented here are powerful enough to improve the F-measure of a NER system. As we expected, adding several morphological features to the system significantly increases the performance for Hungarian, and also makes it slightly better for English.

6.4 Syntactic Information

In this section, we deal with features concerning sentence structure. In the previous section, we have already added some pieces of information about the neighbourhood of the inspected token to serve as rough syntactic features. However, there are more sophisticated ways of dealing with syntax.

Currently, most NE tagged corpora contain chunking information. Chunks are the output of shallow parsing, which gives an analysis of a sentence by identifying its constituents. Chunks are black boxes in the sense that their internal structure is not specified. Moreover, shallow parsing does not provide any information about the syntactic or semantic role of the chunks in a sentence. Chunk tags are similar to NE tags: they have start and end positions and also the category the chunk belongs to (e.g. noun phrase (NP), prepositional phrase (PP)).

The Szeged NER corpus in its original form does not contain syntactic information; however, since it is a part of the Szeged Treebank, it was possible to obtain gold chunk tags. Two versions of the mapping were generated. First, we focused only on baseNPs, which do not contain another NP. The second version contains complete chunking information on maximal length NPs (maxNPs) [Bourigault, 1992]. In the CoNLL-2003 dataset, besides maxNPs, other phrases are also labelled. We converted the original `BI` labelling format to the `BIE1` format which is also used for NE labels (cf. Subsection 5.3.2).

Since we deal with all kinds of sentence features in this section, features of sentence start and end position are also applied here. The features are as follows:

`sentstart`: if the token is in sentence starting position, it gets the feature `sentstart=1`;

`sentend`: if it is in sentence ending position, it gets the feature `sentend=1`;

`chunktag`: the whole chunk tag, e.g. *B-NP*;

`chunktype`: the type of the chunk, e.g. *NP*;

`chunkpart`: the part of the tag which is indicating its position in a chunk, e.g. *B*;

`NpPart`: if it is a part of a NP, it gets the feature `nppart=1`;

`parsePatts`: pattern of chunk labels in the sentence in 5 and 7 words windows.

Table 6.6 shows the results. One of the most interesting findings is that the features indicating sentence start and end position do not improve F-measure in either Hungarian or English. Since they are generally used for NER (e.g. Mayfield et al. [2003]; Chieu and Ng [2003]), in a second trial

they are added together to the system, but this results in an even lower performance, so these features are removed from both the Hungarian and the English system.

feature	**Hungarian baseNP**	**Hungarian maxNP**	**English**
best so far	97.14	97.14	88.02
+sentstart	96.53	-	86.41
+sentend	96.79	-	87.98
+sentstart+sentend	96.43	-	87.74
+chunktag	96.87	96.35	87.75
+chunktype	96.09	96.35	*88.18*
+chunkpart	96.87	95.99	87.23
+getNpPart	96.96	96.43	87.76
+parsePatts			
2rad	96.79	96.26	87.45
3rad	96.61	95.62	87.10

Table 6.6: Results of adding syntactic features one by one to the system.

As for the English data, `chunktype` is the only syntactic feature that improves performance. This can be due to the fact that NEs are generally parts of noun phrases, which can be preceded or followed by other phrases, typically PPs or verbal phrases (VPs). The `parsepatt` feature, whose function is to recognize such typical patterns, is proved to be unnecessary, since every token gets the `chunktype` features of neighbouring tokens as well.

The information coded by the `chunkpart` feature, i.e. recording the position of the token in a phrase is proved not to be useful, since the boundaries of NPs and NEs are seldom the same. Here is an example that illustrates this phenomenon:

token	**POS**	**lemma**	**chunk**	**NE**
to	TO	to	1-PP	O
the	DT	the	B-NP	O
European	NNP	European	I-NP	B-ORG
Union	NNP	union	E-NP	E-ORG

As for Hungarian, neither baseNP nor maxNP features improve F-measure. The example above also fits for Hungarian NPs, since articles

are usually not included in NEs. In contrast with the English results, the `chunktype` feature does not meet our expectations.

The low performance of syntactic features can be caused by the fact that our system has already been expanded with a quite large number of features coding a wide range of information. Pieces of syntactic information seem less important for NER than casing features, digit patterns, or morphological information. Therefore, chunking and NER are sometimes performed in reverse order in NLP applications. Identifying language units as NEs and recognizing their boundaries in the text can help when trying to find chunks. For example, Osenova and Kolkovska [2002] combines the two tasks: NER is assumed prior to the stage of NP chunking. Based on the problem of mismatching boundaries of NPs and NEs, a model which allows joint analysis could be the real solution (for more details, see Section 6.7).

6.5 List Lookup Features

List lookup features are of special importance in NER, as they are used by almost all systems. List inclusion is a way to express the relation 'is a' (e.g. *Budapest is a city*). The assumption behind using this kind of features is that if a token is an element of a list of cities, then the probability of this token to be a city is high.

Several kinds of lists are used for NER. First, using general lists containing common nouns, function words, capitalized nouns or common abbreviations has proven useful in particular for the disambiguation of capitalized words in ambiguous position, e.g. in sentence start and end positions. Mikheev [1999] reports that more than 20% of NEs are ambiguous with common nouns in a corpus built from news of The New York Times. For similar reasons, Chieu and Ng [2003] use a list of lower case words that occur inside names (e.g. *van der*, *of*).

Second, lists of entity cues such as typical endings of organization names, person titles, prefixes and typical location words are also widely used. Some of them are internal evidence of NEs (cf. Subsection 5.2.2). For example, knowing that *Associates*, *Inc.*, and *Corp.* are frequently used in organization names could lead to the recognition of *Barrington Research Associates Inc.* and *AMR Corp.* [Gaizauskas et al., 1995]. Similarly, person and location names are often parts of organization names (e.g. *Lehman Brothers International, Association for Relations Across the Taiwan Straits*), so they are good indicators of organization names [Wolinski et al., 1995]. Others serve as external evidence (cf. Subsection 5.2.3) such as name prefixes

(e.g. *Mr. Jones*) and typical prepositions of location names (e.g. *to London*) [Borthwick, 1999].

Third, lists of NEs that belong to a certain NE type are used most frequently. In the rest of this section, the use of this feature type is discussed more thoroughly.

Most approaches implicitly require candidate words to exactly match an element of a list. However, one may want to allow some flexibility in the match conditions. There are several lookup strategies used in NER. First, words can be lemmatized and only lemmas matched to list elements. For this purpose, the guesser functionality of a morphological analyser can also be used, as we do when applying morphological features (cf. Section 6.3). More sophisticated methods can also be applied, as in the case of our rule-based system (cf. Subsection 5.2.1) and in the case of recognizing metonymic NEs (cf. Subsection 3.3.2).

Second, words can be fuzzy-matched against the list using some kind of metric that measures string distance. This allows capturing small lexical variations in words that are not necessarily inflectional. For example, Tsuruoka and Tsujii [2003] calculate edit distance between spelling variations of protein names in biomedical texts, while Cohen and Sarawagi [2004] use the Jaro-Winkler distance metric to correct mismatches.

6.5.1 The Effects of Gazetteer List Size

One might think that NER can be performed by using lists of person, place and organization names alone, but this is not the case. It is not feasible to list all names, since new companies are formed all the time, and new persons are born, receiving new names. In addition, names can occur in variations: Frederick Flintstone can be mentioned as *Frederick*, *Fred*, *Freddy*, all of which can also be combined with the last name. These variations would have to be listed as well.

Even if it was possible to list all names, there would still be the problem of overlaps between lists, which is caused by the fact that a wide range of names refer to more than one object in the world (cf. Subsection 2.3.1). Moreover, complex NEs can include common nouns or function words. For example, *People's Daily* contains a common noun, a possessive marking and an adverb. If this name is included in a list of organization names, a feature like `isPartOfOrg=1` can be assigned to every mention of its individual words in a text.

In 1998, Cucchiarelli et al. [1998] reported that one of the bottlenecks in designing NER systems is the limited availability of large gazetteers,

particularly for languages other than English. Their system relies on gazetteers of common proper nouns and a set of heuristic rules, similar to our rule-based system described in Subsection 5.2.1. As explained in Subsection 5.2.4, rule-based systems highly depend on the size of gazetteer lists.

However, the situation has changed since the 1990s. Currently, most NER systems use some kind of machine learning algorithm based on probabilities calculated from training data. The fact that we built NER systems for Hungarian and English with F-measures of 97.14% and 88.02% respectively, using only surface features, digit patterns, and morphological information, and without the help of external lists, proves that statistical systems do not rely on gazetteers as much as rule-based systems do. In addition, large amounts of NEs are currently available via the web. Online databases and collaboratively generated resources such as Wikipedia, DBpedia and Freebase open the door to the extraction of large lists of several types of names.

When building an NLP system, finding the balance between precision and recall is one of the most essential requirements. Some applications concentrate on precision, while in others, once a minimum of precision is assured, improvements are dominated by recall issues. Recall is impacted most heavily by OOV effects, and OOV effects themselves are an almost direct function of the lists used in the system. Thus, the best way to improve the performance of a system is to expand the lists, so as to address the leading cause of recall errors. The impact of OOV words on recall can to some extent be mitigated by synonym-based techniques.

To illustrate the effects of standard mitigation techniques on precision and recall, we now make a slight detour and briefly describe a method we used for finding metaphorical expressions by means of different kinds of lists [Babarczy et al., 2010a,b; Babarczy and Simon, 2012]. For the automatic identification of metaphors, we searched the corpus for sentences containing one or more words characterising the source domain and one or more words representing the target domain of a given conceptual metaphor. Three different methods of compiling the word lists were tested: a) word association experiment, b) dictionary of synonyms, and c) reference corpus. The first method is based on the assumption that the expressions people associate with a key word for the source domain and a key word for the target domain can provide a lexical profile for a given metaphor type. Word associations were collected in an online experiment. For the second method, the word lists obtained from the association experiment were expanded with the synonyms listed for the association words in a Hungarian word thesaurus. Compared to the association

list, the size of the word lists substantially increased (see Table 6.7). For the third method, word lists for each source and target domain were extracted from a manually annotated corpus. Based on the three sets of word lists, the test corpus was automatically annotated producing three files in which the sentences were marked by tags showing the type of conceptual metaphor the system identified. Each of the three annotations was then verified manually.

words ↓ / method →	association	synonyms	corpus-based
source domain	1239	6348	126
target domain	674	5094	120

Table 6.7: Number of words in lists compiled by the three methods.

Table 6.8 shows the results of the three methods. The most important findings are that when the association word lists were expanded with synonyms, recall slightly improved, but only at the cost of a decline in precision. The corpus-based method, where the appropriate candidates were accurately extracted by hand, was clearly the most successful of the three strategies. (The values are very low, which indicates that our initial hypothesis – that the co-occurrence of psycholinguistically typical source domain and target domain words in a sentence is a good predictor of metaphoricity – receives no empirical support.)

method	recall (%)	precision (%)	F-measure (%)
association	3.8	7.5	5.6
synonyms	18.1	4.5	11.3
corpus-based	31.3	55.4	43.3

Table 6.8: Results of the three methods.

Turning back to the NER task: for proper names, similar synonym-based mitigation techniques break down. Based on the results presented above, we can hypothesize that expanding the gazetteer lists will result in higher recall at the cost of a decline in precision, and that shorter, but accurately selected lists will improve both precision and recall. Related works on the effects of gazetteer list size in NER also confirm our hypothesis. Morgan et al. [1995], participants of the MUC-6 competition, report that gazetteers provided by the organizers were not used in their

system due to their limited effect on performance. Krupka and Hausman [1998] present lexicon size experiments, where the basic gazetteers contain 110,000 names, which were reduced to 25,000 and 9,000 names, while system performance did not degrade much (from 91.60% to 91.45% and 89.13%, respectively). Moreover, they also show that the addition of an extra 42 entries to the gazetteers improves performance dramatically on MUC-7 datasets. Based on these previous experiments, Mikheev et al. [1999] ask the questions: how important are gazetteers? is their size important? if gazetteers are important but their size is not, then what are the criteria for building gazetteers? They hypothesize and confirm empirically that using relatively small gazetteers of well-known names, rather than large gazetteers of low-frequency names, is sufficient. Indeed, while best results do come from the run with full gazetteers, the run with limited gazetteers decreases precision and recall only with 1–4 percents.

6.5.2 Experiments

To get answers to Mikheevs' questions, we carried out several experiments on gazetteer list size for both languages. We built lists of different sizes of every type of names from different sources. In contrast to the feature adding steps so far, each list was added separately to the system, thus the longer lists do not contain the shorter ones and their effect was measured separately. Since we are interested in the effects of list expansion on precision and recall, here we also provide these figures.

We only include features which are activated when the token is in the dictionary. This model is roughly equivalent to a model containing features indicating that a token is not in the dictionary [Borthwick, 1999]. All experiments were run with the same overall parameters as the previous ones, using the feature combination that proved to be the best so far.

Following the standard method, we first used lists extracted from the corresponding training sets, hoping to tune the system to the kinds of NEs that occur in the particular genre of text (here, newswire). Since training corpora are restricted in size, these gazetteers do not contain many names. In the next experiments, we used longer and longer lists aggregated mostly from the web. For both languages, we also used lists extracted from our Wikipedia corpora (cf. Section 4.3). Finally, we manually created small and accurately selected name lists, and measured the system's performance with these limited gazetteers.

To compile lists for English, we downloaded several kinds of NEs from

the CIA Factbook[1]: names of countries, capitals, cities; languages, nationalities, religions; parties; and party persons, resulting in gazetteers that contain ca. 10,000 names altogether.

For English names, we used Freebase[2], which is an open repository of structured data of almost 23 million entities at the time of writing. Data in Freebase comes from a variety of data sources. Some of them are automatically loaded from a wide range of websites, others are manually added by the Freebase community. We used this large data repository as a list of entities, extracting several types of NEs from amusement park areas to TV characters, and mapping them to CoNLL name categories. Our Freebase lists contain more than 6 million names altogether.

To answer the question of whether gazetteer size is important or not, we compiled lists of three different sizes: from the CIA Factbook, from the English Wikipedia, and from Freebase. In the next step, we seek answers to the question of what the criteria are for building gazetteers. If one wants to base gazetteer building on linguistic criteria, frequency data should be used. For this reason, all three lists described above were merged for each NE type, then occurrences of each name were counted. First, we used only names whose frequency is above 100. Afterwards, the first n names, i.e. the n most frequent names were included in the dictionaries, where n is set to higher and higher values (100, 1,000, 10,000, 100,000).

In the last experiment, we applied an extra-linguistic criterion: as Mikheev et al. [1999] assert, using well-known names are useful for NER. Since the corpora include mostly business newswire, we selected the richest cities and the countries and continents they are located in from Wikipedia, the world's biggest companies according to the Fortune 500 Global list, the richest men in the world who are on the Forbes list of billionares, and the most widely used languages according to the Ethnologue database. These lists contain altogether 903 names, and in contrast to the large lists above, they were cleaned manually. These lists are intended to be small, but accurately compiled gazetteers, which we suppose will improve precision.

Table 6.9 shows results of gazetteer list size experiments on the English data. It can be clearly seen that running the system with different size gazetteers does not change the performance substantially. Precision and recall values are balanced, and F-measures vary in a 1–2% range. The only exception is the case when we used lists extracted from the CoNLL training data. Applying them causes an effect which is exactly the inverse of

[1] `https://www.cia.gov/library/publications/the-world-factbook/`
[2] `http://www.freebase.com/`

lists	entries(#)	precision(%)	recall(%)	F-measure(%)
best so far	0	88.57	87.80	88.18
CoNLL train lists	8,214	93.68	78.86	85.64
CIA Factbook	9,885	89.63	88.14	88.88
enwiki	71,011	89.59	88.61	89.09
Freebase	6,339,915	89.21	87.83	88.52
freq > 100	310	88.10	87.21	87.65
n=100	400	88.14	87.31	87.72
n=1,000	4,000	88.34	87.70	88.02
n=10,000	40,000	89.35	88.22	88.78
n=100,000	400,000	89.88	88.37	*89.12*
by hand	903	88.57	88.05	88.31

Table 6.9: Results of gazetteer size experiments on the English dataset.

what we expected: precision increases by 5%, while recall decreases by 9%. Some of the CoNLL-2003 participants (e.g. Carreras et al. [2003]) report that these gazetteers did not help the recognition of NEs, and were therefore not used. Klein et al. [2003] suggests an explanation: since these lists are built from the training data, they do not increase coverage, and provide only a flat distribution of name phrases whose empirical distributions are spiked.

As the results clearly show, building larger and larger lists does not improve the performance significantly. Using manually compiled short lists results in a similar F-measure as using large Freebase lists, the difference is only 0.21%. Small, but accurately selected gazetteers were supposed to improve precision, but it is left unchanged, while recall somewhat increases. Compiling gazetteers based on frequency appears to be a useful method, and F-measure increases with the number of names taken into account. Indeed, frequency-based lists with the highest n give the best result.

We used similar methods for compiling Hungarian lists, with slight differences. We also built gazetteers from the training set of the Szeged NER corpus, and also extracted names from the Hungarian Wikipedia corpus. Since a Freebase-like repository does not exist for Hungarian, we had to collect names from several sources on the web. We aggregated lists of names of Hungarian towns, streets and other locations from official websites of the post and other offices. We also built a list of common suffixes

that typically occur after place names (e.g. *utca* ('street'), *állomás* ('station')). For the `MISC` category, we compiled a list of Hungarian awards. For organizations, we used the list of the names of all Hungarian companies provided by the Hungarian Justice Department. The exhaustive list of all official Hungarian given names was also used, as well as several name prefixes and common nouns marking rank. For all NE types, we also used the gazetteers compiled for the original `hunner` system (cf. Subsection 5.3.2). Putting all of them together, we obtained lists containing more than 900,000 Hungarian names altogether.

Name occurrences were counted to obtain frequency-based lists for Hungarian. The method was similar to that applied for English, but resulted in lists where the frequency of the first elements was equal to the number of lists collected from different sources, so the frequency was not a useful criterion in the case of Hungarian gazetteers. For this reason, we did not conduct experiments with the Hungarian data using larger and larger gazetteers based on frequency counts.

We now had three lists of different sizes, extracted from the Szeged NER training corpus, the Hungarian Wikipedia corpus, and the web, respectively. Similarly to the English experiments, we also compiled small lists by hand, which contain the Hungarian names of countries and their capitals, business newpapers, stock market indexes, Hungary's 20 biggest companies, and the most frequent Hungarian first and last names. These lists include 702 names altogether, and were cleaned manually.

lists	**entries(#)**	**precision(%)**	**recall(%)**	**F-measure(%)**
best so far	0	97.84	96.45	97.14
Szeged train lists	13,579	97.87	97.87	*97.87*
huwiki	68,212	96.77	95.74	96.26
web lists	907,396	96.96	96.10	96.53
by hand	702	96.95	95.92	96.43

Table 6.10: Results of gazetteer size experiments on the Hungarian dataset.

As the results of experiments on the Hungarian dataset show (see Table 6.10), applying lists of different sizes causes system performance to vary in a small range. Using gazetteers extracted from training data causes more than 1% improvement in recall and only a non-significant increase in precision, resulting in the best F-measure. These results clearly show that applying larger and larger dictionaries collected from several sources

does not significantly improve the system's performance.

We compared the performance of a maximum entropy NER system under various entity list size conditions, ranging from a couple of hundred to several million entries, and conclude that entity list size has only moderate impact on statistical NER systems. If large entity lists are available, we can use them, but their lack does not cause invincible difficulties in the development of NER systems [Kornai and Thompson, 2005].

6.6 Evaluation

After measuring the power of features on development datasets, we have to check our findings on the test set. For this reason, we first ran our system on the corresponding test sets with the feature combination being the best so far. After that, we measured the effect of each major feature category by removing them one by one. We always removed one feature category, while others remained the same.

features	**Hungarian**	**English**
best on devel	97.87	89.12
on test	95.41	84.90
-lex	94.77	82.49
-syn	95.41	*85.22*
-morph	95.08	84.34
-digit	95.29	84.79
-casing	*96.10*	84.90
-string	95.37	72.31

Table 6.11: Results of several feature combinations on test datasets.

Table 6.11 shows the results of several feature combinations evaluated on the test datasets. For both languages, it is true that the evaluation on the corresponding test set results in a 3–4% decline in F-measure, compared to the figures achieved on the development set. This is due to the fact that the genre of texts in the datasets are slightly different: the CoNLL test set contains more sports-related news, and the Szeged NER corpus test set does not contain as much stock market news as the development set.

Most of our expectations based on results of feature engineering on the development set are confirmed. Not using lexicon features does not cause significant change in the overall F-measure. Removing syntactic features

from the English system results in a higher performance, which validates our statement that chunking information is not necessary for NER. (Syntactic features were not added to the Hungarian system, so performance remains the same.) Similarly to the results of the original `hunner` system (cf. Subsection 5.3.2), morphological features do not have significant effect, neither do digit patterns. It is interesting that removing the Boolean-valued features providing casing information improves the performance in the case of Hungarian, and does not decrease it in English. Quite surprisingly, removing the string-valued features, which caused the largest decline in the original `hunner` system, has significantly less effect on the Hungarian system, while the English system breaks down without these features.

Comparing these results to those obtained using the external knowledge of Wikipedia (95.48% for Hungarian, and 86.34% for English; see Subsection 4.3.4 for details), we can conclude that using such external resources and a smaller number of features may improve the performance of a NE tagger.

6.7 Conclusion

Having experimented with most of the features generally used in NER, we can conclude that for a supervised NER system, some of the most simple features, the string-valued features related to the character makeup of words are the strongest. Quite counterintuitively, features indicating casing information and sentence starting position do not improve the performance. Features based on external language processing tools such as morphological analysers and chunkers are not necessary for finding NEs in texts.

As for the effects of gazetteer list size, we can conclude that in a statistical NER system, gazetteers are not as important as in rule-based systems. Adding larger and larger lists to the system does not improve the overall F-measure significantly. When such lists are available, there is no reason not to use them, and applying frequency data for creating better dictionaries can be useful, but these techniques are not essential for building a state-of-the-art NER system.

In this chapter, we applied most features traditionally used in NER. However, it is not an exhaustive study, as there are other features which were not included in our system. For example, several semantic features are also widely used, requiring external resources such as WordNet and Levin's verb classes (cf. Subsection 3.3.2). Using Wikipedia, DBpedia and

other community generated sources of external knowledge for improving the performance of NER systems is also an emerging field (cf. Subsection 4.3.1). Another way of improving the performance of a NE tagger is using the tags emitted by other NER systems as features, as we did for the evaluation of our Wikipedia corpora (cf. Subsection 4.3.4).

As mentioned previously (cf. Subsection 5.2.1 and Section 6.4), NER and other tasks realizing language processing on several linguistic levels are interfering. This raises the question of what kind of language processing model to develop.

From the *cognitive* point of view, this question can be transformed to that of how modular the language system is. A module is a set of processes: it converts an input to an output, and is a black box to other modules, since the processes inside are independent of processes outside. Models in which language processing occurs in this way are called *autonomous*. The opposing view is that processing is *interactive*. Interaction involves the influence of processing levels on each other, which raises two more questions. First, are the processing stages discrete or do they overlap? In a discrete model, a level of processing can only begin when the previous one has finished. In a cascade model [McClelland, 1979], information is allowed to flow from one level to the following even before the first process is completed. If the stages overlap, then multiple candidates may become activated at the lower level of processing. The second question of interaction is whether there is a reverse flow of information from a level to a previous one [Harley, 2001].

From the point of view of NER, our system presented here can be viewed as an interactive model in the sense that pieces of surface, morphological, and syntactic information are all provided to the system, and these interfere and compete to solve the task of identifying NEs in the text. For computational reasons, POS tagging, chunking, and NER are defined as discrete processing stages, but actually our NE tagger does not function as a modular system. Moreover, it can be used as a cascade model by the assembly of POS tagging, NP chunking, and NE tagging subsystems.

Chapter 7

Conclusions and Future Directions

The first question the NER task raises is what kind of linguistic units are to be considered NEs. In Chapter 2, we gave an overview of the definition of proper names from the point of view of philosophy and linguistics. We concluded that it is still a challenging question, but there are a few statements which can be used as pillars of defining what to annotate as NEs. If we insist tagging proper names only, we have to restrict the domain of taggables to linguistic units which have unique reference in all possible worlds, thus being rigid designators; which are arbitrary linguistic units whose only semantic implication is the fact of naming; and which are indivisible and non-compositional.

These requirements can serve as the foundation for the definition of every kind of NE, but they must be loosened to allow tagging other important groups of linguistic structures such as relative time expressions. Moreover, there are a quite large number of linguistic units which are difficult to categorize and vary across languages, such as names of nationalitites, languages, days, or brands.

Therefore, a universal definition of NEs that can be applied to all types and languages cannot be given based on the classic Aristotelian view on classification, which states that there must be a *differentia specifica* which allows something to be the member of a group, and excludes others. For the purposes of NER, the prototype theory is more plausible. According to this approach, linguistic units can be seen as elements of a range from the most prototypical to non-prototypical categories. Psycholinguistic experiments (e.g. Kobeleva [2008]) and corpus-based studies (e.g. Tse [2005]) also confirm that person names constitute the core of proper names. Location names occupy an intermediate position, while names of events and

artefacts are considered the least prototypical, i.e. peripheral members of proper names. It is an interesting supplementary observation that the most prototypical names, i.e. person names have been studied from the very beginning of linguistics, and they have been postulated as proper names in the first systematic grammars.

According to this approach, one of the elementary steps of building a NE tagged corpus is creating a continuum of NEs ranging from prototypical to non-prototypical categories, which is an interesting future research direction in Hungarian NER. Finally, the goal of the NER application will restrict the range of linguistic units to be taken into account.

NEs are ambiguous referential elements of discourse, since they are likely to occur in metonymies. Metonymy is a reference shift: we use a name not to refer to its primary reference, but to a related one, i.e. a contextual reference. In linguistics, metonymy is often postulated as sense extension, but because of the meaninglessness of the proper names (cf. Subsection 2.3.2), using the term 'reference shift' is much more suitable.

Since the conceptual mapping between primary and contextual reference is not linked to particular linguistic forms, metonymy is known to pose a difficult task for both human annotators and NLP applications. However, using some surface and syntactic information, and applying several semantic generalization methods lead to improvement in resolving metonymies, which is also suggested by the fact that these features are used by several independent research teams (e.g. Nastase and Strube [2009]; Ferraro [2011]; Judea et al. [2012]). We presented a supervised system, which achieved the best overall results in the SemEval-2007 metonymy resolution task (cf. Chapter 3). Based on the results of our system, we concluded that the main borderline does not lie between conventional and unconventional metonymies, but rather between literal and metonymic usage.

Recognizing metonymic NEs is of key importance in several NLP tasks, such as MT, IR, or anaphora resolution. For this reason, an annotation approach is required that offers the possibility of handling metonimicity at higher processing levels, while providing interoperability between various annotation schemes. This can be achieved by applying the combination of the Tag for Meaning and Tag for Tagging rules, i.e. annotating metonymic NEs with tags which provide information about the primary reference as well as the contextual reference. Such a combination has been applied in case of the HunNer and the Criminal NE corpora. The latter can serve as a training corpus for applying the GYDER system for Hungarian, which is an interesting future direction.

Machine learning algorithms typically learn their parameters from cor-

pora, and systems are evaluated by comparing their output to another part of the corpus, or to another corpus. The corpora which are manually annotated with linguistic information following the rules of some annotation guidelines are gold standard corpora. However, gold standard corpora in the field of NER are highly domain-specific, use different tagsets, and are restricted in size. Manually annotating large amounts of text is a time-consuming, highly skilled, and delicate job, but large, accurately annotated corpora are essential for building robust supervised machine learning NER systems. Therefore, reducing the annotation cost is a key challenge.

An approach to this issue is to generate resources automatically, which can be done by various means, e.g. by applying NLP tools that are accurate enough to allow automatic annotation, or merging existing gold standard datasets. In the latter case, researchers are faced with the problem of having to combine various tagsets and annotation schemes. Another approach is to use collaborative annotation or collaboratively constructed resources, such as Wikipedia or DBpedia. In Section 4.3, we presented a method which combines these approaches by automatically generating NE tagged corpora from Wikipedia.

Automatically generated or silver standard corpora provide an alternative solution which is intended to serve as an approximation of gold standard corpora. Such corpora are highly useful for improving the performance of NER systems in several ways, as shown in Subsection 4.3.4: (a) for less resourced languages, they can serve as training corpora *in lieu* of gold standard datasets; (b) they can serve as supplementary or independent training sets for domains differing from newswire; (c) they can be the source of large entity lists, and (d) feature extraction.

Besides reducing the annotation cost of corpus building, several current trends concerning the NER task emerge from our overview (Chapter 4). Researchers attempting to evaluate their systems across different domains are faced with the fact that cross-domain evaluation results in low F-measure. Thus, current efforts are directed to achieve robust performance across domains, which still remains a problem and needs further investigation.

Another trend in NER research is scaling up to fine-grained entity types. Classic gold standard datasets use coarse-grained NE hierarchies, taking into account only the three main classes of names (`PER`, `ORG`, `LOC`) and certain other types depending on the applied annotation scheme. Fine-grained NE hierarchies also exist (e.g. Sekine's extended hierarchy [Sekine et al., 2002] or the tagset applied in the BBN corpus [Weischedel and Brunstein, 2005]), but when used for evaluation, they have to be

mapped to the classic coarse-grained typology, which is far from trivial. As shown in Section 4.3, using continuously growing, collaboratively constructed resources creates the possibility of building corpora with even more fine-grained NE hierarchies, which can also provide interoperability between various tagsets.

The NER task, similarly to other NLP tasks, can be approached in two main ways: by applying hand-crafted rules, or by statistical machine learning techniques, called rationalist and empiricist methodologies based on their roots in philosophy. Since the 1950s and the 1990s, mainly rationalist methods have been applied in NLP, but empirical methods returned in the 1990s, and have since become the most widely applied techniques.

The rationalist methodology has several shortcomings. Manually developing a rule-based system and handling a large number of rules is quite difficult, requiring a great deal of domain-specific knowledge engineering. In addition, these systems are brittle and not portable between different domains or tasks. Empirical methods, on the other hand, offer potential solutions to several problems in NLP, e.g. knowledge acquisition by means of automatic learning techniques, coverage by means of large amounts of data, robustness by means of frequency-based algorithms, and extensibility by means of portable systems.

Despite the fact that gold standard corpora used for training and testing are usually cleaned, empirical methods still offer more robustness than rule-based systems do, in that they can be ported to new datasets or new domains at the expense of some performance loss, but without the need to rewrite the whole system. Although robustness and extensibility are still not properly solved, as illustrated in Section 4.1 and in Subsection 4.2.2, respectively, using empirical methods clearly results in higher performance.

As the strengths and weaknesses of statistical and rule-based systems tend to be complementary, current research attempts to deal with hybrid solutions that combine the two methodologies. However, these approaches have so far been less successful in industrial applications than in the research lab.

The current dominant technique used in the field of NER is supervised learning. Its disadvantage is that it requires large amounts of previously annotated data (cf. Chapter 4 and 5), so one might say that the human labour of creating rules has only been shifted to that of building corpora. However, there is a quite new, emerging field of NLP and of NER in particular, which involves using unsupervised or semi-supervised techniques. In these cases, human labour has also been shifted to constructing seed examples and/or embedding heuristics in the system. Unsupervised learning is a field where significant improvements can be made in the future.

Another interesting future direction is developing cascade models of several NLP tasks overlapping and allowing information flow between stages. Such attempts were made by Móra and Vincze [2012] at joint POS tagging and NER, and by Finkel and Manning [2009] at joint parsing and NER. However, this approach is still in its infancy and needs further investigation.

Defining features, which are descriptors or characteristic attributes of datapoints in the text, is a manual undertaking, similar to coding patterns for a rule-based system. However, in the case of statistical methods, the linguist does not furnish information about the power of the feature, which has to be measured on real data before inclusion into the system. To measure the strength of features, we used our maximum entropy NER system (see Subsection 5.3.2), and made several experiments with adding new features to it one by one.

After trying out most features generally used in NER, we concluded that for a supervised NER system, the most simple features, i.e. the string-valued features related to the character makeup of words are the strongest ones. Quite counterintuitively, features indicating casing information and sentence starting position do not improve performance. Features based on external language processing tools such as morphological analysers and chunkers do not seem necessary for finding NEs in texts. Therefore, our system does not serve as a modularized language processing model (cf. Section 6.7).

Bibliography

Abney, S. (1996). Statistical Methods and Linguistics. In Klavans, J. and Resnik, P., editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 1–26. MIT Press.

ACE (2008). *ACE (Automatic Content Extraction) English Annotation Guidelines for Entities. Version 6.6*. Linguistic Data Consortium. http://projects.ldc.upenn.edu/ace/docs/English-Entities-Guidelines_v6.6.pdf.

Anderson, J. M. (2007). *The Grammar of Names*. Oxford University Press.

Armstrong-Warwick, S. (1993). Preface. *Computational Linguistics*, 19(1):iii–iv.

Artstein, R. and Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4).

Atserias, J., Zaragoza, H., Ciaramita, M., and Attardi, G. (2008). Semantically Annotated Snapshot of the English Wikipedia. In *Proceedings of LREC 2008*.

Babarczy, A., Bencze, I., Fekete, I., and Simon, E. (2010a). A metaforikus nyelvhasználat egy korpuszalapú elemzése. In Tanács, A. and Vincze, V., editors, *VII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 145–156, Szeged.

Babarczy, A., Bencze, I., Fekete, I., and Simon, E. (2010b). The Automatic Identification of Conceptual Metaphors in Hungarian Texts: A Corpus-based Analysis. In Bel, N., Daille, B., and Vasiljevs, A., editors, *Proceedings of the LREC 2010 Workshop on Methods for the automatic acquisition of Language Resources and their evaluation methods*, pages 31–36, Malta.

Babarczy, A., Serény, A., and Simon, E. (2009). Magyar igei vonzatkeretek gépi tanulása. In Tanács, A., Szauter, D., and Vincze, V., editors, *VI.*

Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2009), pages 333–342, Szeged. SZTE.

Babarczy, A. and Simon, E. (2012). A fogalmi metaforák és a szövegstatisztika szerepe a metaforák felismerésében. In Prószéky, G. and Váradi, T., editors, *Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások*, pages 223–241. Akadémiai Kiadó, Budapest.

Bahl, L. R. and Mercer, R. L. (1976). Part-of-Speech Assignment by a Statistical Decision Algorithm. In *Abstracts of papers from the IEEE International Symposium on Information Theory*, pages 88–89, Washington. IEEE Computer Society.

Benajiba, Y., Diab, M., and Rosso, P. (2008). Arabic Named Entity Recognition using Optimized Feature Sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 284–293, Stroudsburg, PA, USA. Association for Computational Linguistics.

Biber, D. (1993). Representativeness of Corpus Design. *Literary and Linguistic Computing*, 8(4).

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a High-Performance Learning Name-finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 194–201.

Bikel, D. M., Schwartz, R., and Weischedel, R. M. (1999). An Algorithm that Learns What's in a Name. *Machine Learning*, 34:211–231.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). DBpedia – A crystallization point for the Web of Data. *Web Semantics*, 7(3):154–165.

Borthwick, A. (1999). *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University.

Bourigault, D. (1992). Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 977–981.

Brill, E. and Mooney, R. J. (1997). An Overview of Empirical Natural Language Processing. *AI Magazine*, 18(4):13–24.

Brill, E. and Moore, R. C. (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

Brun, C., Ehrmann, M., and Jacquet, G. (2007). XRCE-M: A Hybrid System for Named Entity Metonymy Resolution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 488–491, Prague, Czech Republic. Association for Computational Linguistics.

Brunstein, A. (2002). *Annotation Guidelines for Answer Types*. BBN Technologies. http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html.

Bunescu, R. and Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16.

Carreras, X., Màrquez, L., and Padró, L. (2003). A Simple Named Entity Extractor using AdaBoost. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 152–155. Edmonton, Canada.

Chieu, H. L. and Ng, H. T. (2003). Named Entity Recognition with a Maximum Entropy Approach. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 160–163. Edmonton, Canada.

Chinchor, N. (1998a). MUC-7 Named Entity Task Definition Version 3.5. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*.

Chinchor, N. (1998b). Overview of MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*.

Chinchor, N., Brown, E., Ferro, L., and Robinson, P. (1999). *1999 Named Entity Recognition Task Definition Version 1.4*.

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.

Chomsky, N. (1959). A review of B. F. Skinner's Verbal Behavior. *Language*, 35(1):26–58.

Ciaramita, M. and Altun, Y. (2005). Named-entity Recognition in Novel Domains with External Lexical Knowledge. In *Proceedings of the NIPS Workshop on Advances in Structured Learning for Text and Speech Processing*.

Clercq, O. D. and Perez, M. M. (2010). Data Collection and IPR in Multilingual Parallel Corpora. Dutch Parallel Corpus. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and

Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Cohen, W. W. and Sarawagi, S. (2004). Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods. In *Conference on Knowledge Discovery in Data: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98.

Collins, M. (2002). Ranking Algorithms for Named Entity Extraction: Boosting and the Voted Perceptron. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Collins, M. and Singer, Y. (1999). Unsupervised Models for Named Entity Classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Csendes, D., Csirik, J., and Gyimóthy, T. (2004). The Szeged Corpus. A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus. In Hansen-Schirra, S., Oepen, S., and Uszkoreit, H., editors, *COLING 2004 5th International Workshop on Linguistically Interpreted Corpora*, pages 19–22, Geneva, Switzerland. COLING.

Csendes, D., Csirik, J., Gyimóthy, T., and Kocsor, A. (2005). The Szeged Treebank. In *Proceedings of the 8th International Conference, TSD 2005*, pages 123–131, Karlovy Vary, Czech Republic. Springer.

Cucchiarelli, A., Luzi, D., and Velardi, P. (1998). Automatic Semantic Tagging of Unknown Proper Names. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 286–292, Montreal, Quebec, Canada. Association for Computational Linguistics.

Cucchiarelli, A. and Velardi, P. (2001). Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. *Computational Linguistics*, 27(1):123–131.

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Cumming, S. (2012). Names. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Spring 2012 edition.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*.

Curran, J. and Clark, S. (2003). Language Independent NER using a Maximum Entropy Tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167.

Deme, L. (1956). Családneveink alaki sérthetetlenségéről. *Magyar Nyelv*, 52:365–368.

Elsner, M., Charniak, E., and Johnson, M. (2009). Structured Generative Models for Unsupervised Named-Entity Clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172, Boulder, Colorado. Association for Computational Linguistics.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Farkas, R. (2007). Tulajdonnév-felismerés. In Tikk, D., editor, *Szövegbányászat*, pages 90–98. TypoTEX, Budapest.

Farkas, R., Simon, E., Szarvas, Gy., and Varga, D. (2007). GYDER: maxent metonymy resolution. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 161–164, Prague. Association for Computational Linguistics.

Farkas, R., Vincze, V., Nagy, I., Ormándi, R., Szarvas, Gy., and Almási, A. (2008). Web-Based Lemmatisation of Named Entities. In *Proceedings of the 11th international conference on Text, Speech and Dialogue, TSD '08*, pages 53–60, Berlin, Heidelberg. Springer-Verlag.

Fass, D. (1988). Metonymy and Metaphor: What's the Difference? In *Proceedings of the 12th Conference on Computational linguistics – Volume 1*, COLING '88, pages 177–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Ferraro, F. M. O. (2011). Toward Improving the Automated Classification of Metonymy in Text Corpora. Honors Bachelor of Science thesis, Department of Computer Science, University of Rochester, Rochester, NY.

Finkel, J. R. and Manning, C. D. (2009). Joint Parsing and Named Entity Recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334, Stroudsburg, PA, USA. Association for Computational Linguistics.

Frege, G. (2000). Ueber Sinn und Bedeutung (On Sense and Reference). In Stainton, R. J., editor, *Perspectives in the Philosophy of Language – A concise anthology*. Broadview Press.

Gábor, K., Héja, E., Mészáros, Á., and Sass, B. (2003). *Nyílt tokenosztályok reprezentációjának technológiája*. Szeged. IKTA-00037/2002, I. munkaszakasz beszámoló.

Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., and Wilks, Y. (1995). University of Sheffield: Description of the LaSIE System as Used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland.

Gardiner, A. (1957). *The Theory of Proper Names. A Controversial Essay*. Oxford University Press, London.

Gendler Szabó, Z. (2008). Compositionality. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Winter 2008 edition.

Grice, H. P. (1975). Logic and Conversation. In Cole, P. and Morgan, J. L., editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York.

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference – 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 466–471, Kopenhagen.

Halácsy, P., Kornai, A., and Oravecz, Cs. (2007). HunPos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212, Prague, Czech Republic. Association for Computational Linguistics.

Halácsy, P., Kornai, A., and Varga, D. (2005). Morfológiai egyértelműsítés maximum entrópia módszerrel. In *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2005)*, pages 180–189, Szeged.

Harabagiu, S. (1998). Deriving Metonymic Coercions from WordNet. In *Workshop on the Usage of WordNet in Natural Language Processing Systems, COLING ACL*, pages 142–148.

Harley, T. A. (2001). *The Psychology of Language. From Data to Theory*. Psychology Press Ltd., second edition.

Harris, Z. (1951). *Methods in Structural Linguistics*. University of Chicago Press, Chicago.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. (1993). Interpretation as Abduction. *Artificial Intelligence*, 63(1-2):69–142.

Huddleston, R. and Pullum, G. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press, New York.

Hunston, S. (2008). Collection strategies and design decisions. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics. An International Handbook*, pages 154–167. Walter de Gruyter, Berlin.

Judea, A., Nastase, V., and Strube, M. (2012). Concept-based Selectional Preferences and Distributional Representations from Wikipedia Articles. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2985–2990, Istanbul, Turkey. European Language Resources Association (ELRA).

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing. An Introduction to Natural language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Kamei, S. and Wakao, T. (1992). Metonymy: Reassessment, survey of acceptability and its treatment in machine translation systems. In *Proceedings of ACL*, pages 309–311.

Katz, J. J. (1972). *Semantic Theory*. Harper and Row, New York.

Kazama, J. and Torisawa, K. (2007). Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

Kenesei, I., Vago, R. M., and Fenyvesi, A. (2012). *Hungarian*. Descriptive Grammars. Taylor & Francis.

Kiss, T. and Strunk, J. (2006). Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, 32(4).

Klein, D., Smarr, J., Nguyen, H., and Manning, C. D. (2003). Named Entity Recognition with Character-Level Models. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 180–183. Edmonton, Canada.

Kobeleva, P. P. (2008). *The Impact of Unfamiliar Proper Names on ESL Learners' Listening Comprehension*. PhD thesis, Victoria University of Wellington, New Zealand.

Kornai, A. (1994). *On Hungarian morphology*, volume 14. of *Linguistica Series A, Studia et Dissertationes*. Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest.

Kornai, A. and Thompson, B. (2005). Size doesn't matter. Unpublished manuscript.

Kripke, S. (2000). Naming and Necessity. In Stainton, R. J., editor, *Perspectives in the Philosophy of Language – A concise anthology*. Broadview Press.

Krupka, G. R. and Hausman, K. (1998). IsoQuest Inc.: Description of the NetOwl^{TM}Extractor System as Used for MUC-7. In *Proceedings of the 7th Message Understandig Conference (MUC-7)*.

Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Brown University Press, Providence.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. Chicago University Press, London.

Landis, J. R. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Leech, G., Garside, R., and Atwell, E. (1983). The Automatic Grammatical Tagging of the LOB Corpus. *ICAME News*, 7:13–33.

Leveling, J. and Hartrumpf, S. (2006). On Metonymy Recognition for GIR. In *Proceedings of GIR-2006, the 3rd Workshop on Geographical Information Retrieval (hosted by SIGIR 2006)*, Seattle, Washington.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. The University of Chicago Press, Chicago.

Linguistic Data Consortium LCTL Team (2006). *Simple Named Entity Guidelines For Less Commonly Taught Languages. Version 6.5*.

Lukács, Á. (2001). Szabályok és kivételek: a kettős modell érvényessége a magyarban. In Pléh, Cs. and Lukács, Á., editors, *A magyar morfológia pszicholingvisztikája*, pages 119–152. BIP – Osiris Kiadó.

LXMLS (2011). *LxMLS Lab Guide*. Instituto Superior Técnico, Instituto de Telecomunicacoes, and INESC-ID, Lisbon, Portugal. Provided by the organizers of the First Lisbon Machine Learning School: Learning for the Web – LxMLS 2011.

Lüdeling, A. and Kytö, M., editors (2008). *Corpus Linguistics. An International Handbook*. Walter de Gruyter, Berlin.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Markert, K. and Hahn, U. (2002). Understanding Metonymies in Discourse. *Artificial Intelligence*, 135(1/2):145–198.

Markert, K. and Nissim, M. (2002). Metonymy Resolution as a Classification Task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 204–213, Philadelphia. Association for Computational Linguistics.

Markert, K. and Nissim, M. (2007a). Metonymic Proper Names: A Corpus-based Account. In Stefanowitsch, A. and Gries, S. T., editors, *Corpus-Based Approaches to Metaphor and Metonymy*, pages 152–174. Mouton de Gruyter.

Markert, K. and Nissim, M. (2007b). SemEval-2007 Task 08: Metonymy Resolution at SemEval-2007. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 36–41, Prague. Association for Computational Linguistics.

Mayfield, J., McNamee, P., and Piatko, C. (2003). Named Entity Recognition using Hundreds of Thousands of Features. In *Proceedings of CoNLL-2003*, pages 184–187.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., and Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*.

McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598.

McCallum, A. and Li, W. (2003). Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In Daelemans, W. and Osborne, M., editors, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191.

McClelland, J. L. (1979). On the Time Relations of Mental Processes: An Examination of Systems of Processes in Cascade. *Psychological Review*, 86:287–330.

McDonald, D. D. (1996). Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In Boguraev, B. and Pustejovsky, J., editors, *Corpus Processing for Lexical Aquisition*, pages 21–39. MIT Press, Cambridge, MA.

McEnery, T. (2004). Corpus Linguistics. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*, pages 448–463. Oxford University Press, New York.

McEnery, T. and Wilson, A. (2001). *Corpus Linguistics*. Edinburgh University Press, Edinburgh, 2nd edition.

Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining Meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754.

Merchant, R., Okurowski, M. E., and Chinchor, N. (1996). The Multilingual Entity Task (MET) Overview. In *Proceedings of the TIPSTER Text Program: Phase II*, pages 445–447, Vienna, Virginia, USA. Association for Computational Linguistics.

Mikheev, A. (1999). A Knowledge-free Method for Capitalized Word Disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 159–166, College Park, Maryland, USA. Association for Computational Linguistics.

Mikheev, A., Moens, M., and Grover, C. (1999). Named Entity Recognition without Gazetteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway.

Mill, J. S. (2002). *A System of Logic*. University Press of the Pacific, Honolulu.

Miller, D., Schwartz, R., Weischedel, R., and Stone, R. (1999). Named Entity Extraction from Broadcast News. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, Virginia.

Mooney, R. J. (2004). Machine Learning. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*, pages 376–394. Oxford University Press, New York.

Móra, Gy. and Vincze, V. (2012). Joint Part-of-Speech Tagging and Named Entity Recognition Using Factor Graphs. In Sojka, P., Horák, A., Kopecek, I., and Pala, K., editors, *Text Speech and Dialogue. Proceedings of the 15th International Conference, TSD 2012*, pages 232–239, Brno, Czech Republic. Springer.

Morgan, R., Garigliano, R., Callaghan, P., Poria, S., Smith, M., Urbanowicz, A., Collingham, R., Costantino, M., Cooper, C., and The LOLITA Group (1995). University of Durham: Description of the LOLITA system as used in MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland.

Nadeau, D. and Sekine, S. (2007). A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26.

Nadeau, D., Turney, P., and Matwin, S. (2006). Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity. *Advances in Artificial Intelligence*, pages 266–277.

Narayanaswamy, M., Ravikumar, K. E., and Vijay-Shanker, K. (2003). A Biological Named Entity Recognizer. In *Pacific Symposium on Biocomputing*.

Nastase, V. and Strube, M. (2009). Combining Collocations, Lexical and Encyclopedic Knowledge for Metonymy Resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 910–918, Singapore. Association for Computational Linguistics.

Nemeskey, D. M. and Simon, E. (2012). Automatikus korpuszépítés tulajdonnév-felismerés céljára. In Tanács, A. and Vincze, V., editors, *IX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013)*, pages 106–117, Szeged.

Németh, L., Trón, V., Halácsy, P., Kornai, A., Rung, A., and Szakadát, I. (2004). Leveraging the open source ispell codebase for minority language analysis. In *First Steps in Language Documentation for Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, Proceedings of the SALTMIL Workshop at LREC*, pages 56–59.

Nicolae, C., Nicolae, G., and Harabagiu, S. (2007). UTD-HLT-CG: Semantic Architecture for Metonymy Resolution and Classification of Nominal Relations. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 454–459, Prague, Czech Republic. Association for Computational Linguistics.

Nissim, M. and Markert, K. (2005). Learning to buy a Renault and talk to BMW: A supervised approach to conventional metonymy. In *International Workshop on Computational Semantics (IWCS2005)*, Tilburg, Netherlands.

Nothman, J., Curran, J. R., and Murphy, T. (2008). Transforming Wikipedia into Named Entity Training Data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132.

O'Keeffe, A. and McCarthy, M., editors (2010). *The Routledge Handbook of Corpus Linguistics*. Routledge.

Oravecz, Cs., Sass, B., and Simon, E. (2009). Gépi tanulási módszerek ómagyar kori szövegek normalizálására. In Tanács, A., Szauter, D., and Vincze, V., editors, *VI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2009)*, pages 317–324, Szeged. SZTE.

Oravecz, Cs., Sass, B., and Simon, E. (2010). Semi-automatic Normalization of Old Hungarian Codices. In *Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010)*, pages 55–60, Lisbon, Portugal. Faculty of Science, University of Lisbon.

Osenova, P. and Kolkovska, S. (2002). Combining the named-entity recognition task and NP chunking strategy for robust pre-processing. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September*, pages 20–21.

Poibeau, T. (2007). UP13: Knowledge-poor Methods (Sometimes) Perform Poorly. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 418–421, Prague, Czech Republic. Association for Computational Linguistics.

Poibeau, T. and Kosseim, L. (2001). Proper Name Extraction from Non-Journalistic Texts. In *Computational Linguistics in the Netherlands*, pages 144–157.

Pustejovsky, J. (1995). *The Generative Lexicon*. MIT Press, Cambridge, Massachusetts.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quirk, R. and Greenbaum, S. (1980). *A University Grammar of English*. Longman.

Radden, G. and Kövecses, Z. (1999). Towards a Theory of Metonymy. In Panther, K.-U. and Radden, G., editors, *Metonymy in Language and Thought*, pages 17–60. John Benjamins.

Rebrus, P., Kornai, A., and Varga, D. (2012). Egy általános célú morfológiai annotáció. In Prószéky, G. and Váradi, T., editors, *Általános Nyelvészeti Tanulmányok XXIV.*, pages 47–80. Akadémiai Kiadó, Budapest.

Recski, G. and Varga, D. (2010). A Hungarian NP Chunker. *The Odd Yearboook*, 8:87–93.

Reddy, M. J. (1979). The conduit metaphor: A case of frame conflict in our language about language. In Ortony, A., editor, *Metaphor and Thought*, pages 284–310. Cambridge University Press, Cambridge.

Richman, A. E. and Schone, P. (2008). Mining Wiki Resources for Multilingual Named Entity Recognition. In *Proceedings of ACL-08: HLT*, pages 1–9, Columbus, Ohio. Association for Computational Linguistics.

Rindfleish, T., Tanabe, L., and Weinstein, J. (2000). EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature. In *Proceedings of Pacific Symposium on Bioinformatics*, pages 514–525, Hawaii, USA.

Rosch, E. H. (1973). Natural Categories. *Cognitive Psychology*, 4(3):328 – 350.

Russell, B. (2000). Descriptions. In Stainton, R. J., editor, *Perspectives in the Philosophy of Language – A Concise Anthology*. Broadview Press.

Saussure, F. d. (1959). *Course in General Linguistics*. Philosophical Library, New York.

Sekine, S., Sudo, K., and Nobata, C. (2002). Extended named entity hierarchy. In *Proceedings of LREC*, volume 2.

Serény, A., Simon, E., and Babarczy, A. (2009). Automatic acquisition of Hungarian subcategorization frames. In *Proceedings of the 9th International Symposium of Hungarian Researchers on Computational Intelligence*.

Settles, B. (2004). Biomedical Named Entity Recognition using Conditional Random Fields and Rich Feature Sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423.

Shannon, C. E. (1951). Prediction and Entropy of Printed English. *Bell Systems Technical Journal*, 30:50–64.

Simon, E. (2008). Nyelvészeti problémák a tulajdonnév-felismerés területén. In Sinkovics, B., editor, *LingDok 7. Nyelvész-doktoranduszok dolgozatai*, pages 181–196. Szegedi Tudományegyetem Nyelvtudományi Doktori Iskola, Szeged.

Simon, E., Farkas, R., Halácsy, P., Sass, B., Szarvas, Gy., and Varga, D. (2006). A HunNER korpusz. In Alexin, Z. and Csendes, D., editors, *IV. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged.

Simon, E. and Nemeskey, D. M. (2012). Automatically generated NE tagged corpora for English and Hungarian. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 38–46, Jeju, Korea. Association for Computational Linguistics.

Simon, E. and Sass, B. (2012). Nyelvtechnológia és kulturális örökség, avagy korpuszépítés ómagyar kódexekből. In Prószéky, G. and Váradi, T., editors, *Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások*, pages 243–264. Akadémiai Kiadó, Budapest.

Simon, E., Sass, B., and Mittelholcz, I. (2011). Korpuszépítés ómagyar kódexekből. In Tanács, A. and Vincze, V., editors, *VIII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 81–89, Szeged. SZTE.

Simon, E., Serény, A., and Babarczy, A. (2010). Automatic Acquisition of Hungarian Subcategorization Frames. In Bel, N., Daille, B., and Vasiljevs, A., editors, *Proceedings of the LREC 2010 Workshop on Methods for the automatic acquisition of Language Resources and their evaluation methods*, pages 7–11, Malta.

Sinclair, J. (2005). Corpus and Text – Basic Principles. In Wynne, M., editor, *Developing Linguistic Corpora: a Guide to Good Practice*, pages 1–16. Oxbow Books, Oxford.

Stallard, D. (1993). Two kinds of metonymy. In *Proceedings of ACL*, pages 87–94.

Sundheim, B. (1995). MUC-6 Named Entity Task Definition (v2.1). In *Proceedings of the Sixth Message Understanding Conference (MUC6)*.

Svartvik, J., editor (1990). *The London Corpus of Spoken English: Description and Research*, volume 82 of *Lund Studies in English*. Lund University Press.

Szarvas, Gy. (2008). *Feature Engineering for Domain Independent Named Entity Recognition and Biomedical Text Mining Applications*. PhD thesis, University of Szeged.

Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., and Csirik, J. (2006a). A highly accurate Named Entity corpus for Hungarian. In *Electronic Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Szarvas, Gy., Farkas, R., and Kocsor, A. (2006b). A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In *Proceedings of Discovery Science 2006*, pages 267–278. Springer Verlag.

Tanabe, L., Xie, N., Thom, L., Matten, W., and Wilbur, W. (2005). GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC bioinformatics*, 6(Suppl 1):S3.

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In Roth, D. and van den Bosch, A., editors, *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*. Edmonton, Canada.

Toral, A. and Muñoz, R. (2006). A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In *EACL 2006*.

Trón, V., Gyepesi, Gy., Halácsy, P., Kornai, A., Németh, L., and Varga, D. (2005a). Hunmorph: Open Source Word Analysis. In *Proceedings of the ACL Workshop on Software*, pages 77–85, Ann Arbor, Michigan. Association for Computational Linguistics.

Trón, V., Halácsy, P., Rebrus, P., Rung, A., Simon, E., and Vajda, P. (2005b). morphdb.hu: magyar morfológiai nyelvtan és szótári adatbázis. In *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2005)*, pages 169–179, Szeged.

Trón, V., Halácsy, P., Rebrus, P., Rung, A., Vajda, P., and Simon, E. (2006a). Morphdb.hu: Hungarian lexical database and morphological grammar.

In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 1670–1673.

Trón, V., Halácsy, P., Rebrus, P., Rung, A., Vajda, P., and Simon, E. (2006b). Morphdb.hu: Hungarian lexical database and morphological grammar. In S. Nagy, K. and Szakadát, I., editors, *Média és társadalom. Válogatás a Szociológia és Kommunikáció Tanszék Média Oktató és Kutató Központ munkatársainak legújabb munkáiból*, pages 283–290. Műegyetemi Kiadó.

Tse, Grace, Y. W. (2005). *A Corpus-based Study of Proper Names in Present-day English. Aspects of Gradience and Article Usage*. Peter Lang.

Tsuruoka, Y. and Tsujii, J. (2003). Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 41–48, Sapporo, Japan. Association for Computational Linguistics.

Van Langendonck, W. (2007). *Theory and Typology of Proper Names*. Mouton de Gruyter.

Váradi, T. (2002). The Hungarian National Corpus. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 385–389, Las Palmas de Gran Canaria. European Language Resources Association.

Varga, D. and Simon, E. (2006). Magyar nyelvű tulajdonnév-felismerés maximum entrópia módszerrel. In Alexin, Z. and Csendes, D., editors, *IV. Magyar Számítógépes Nyelvészeti Konferencia*, pages 32–38, Szeged.

Varga, D. and Simon, E. (2007). Hungarian named entity recognition with a maximum entropy approach. *Acta Cybernetica*, 18:293–301.

Weischedel, R. and Brunstein, A. (2005). *BBN Pronoun Coreference and Entity Type Corpus*. Linguistic Data Consortium, Philadelphia.

Wolinski, F., Vichot, F., and Dillet, B. (1995). Automatic Processing of Proper Names in Texts. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*.

Xiao, R. (2010). Corpus Creation. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL.

Zhang, L., Pan, Y., and Zhang, T. (2004). Focused Named Entity Recognition using Machine Learning. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 281–288, New York, NY, USA. ACM.

Zhou, G. and Su, J. (2000). Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 473–480.

Zhu, C., Byrd, R., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560.

Zhu, J., Uren, V., and Motta, E. (2005). ESpotter: Adaptive Named Entity Recognition for Web Browsing. In *3rd Conference on Professional Knowledge Management*, pages 518–529.

Zsibrita, J., Vincze, V., and Farkas, R. (2010). Ismeretlen kifejezések és a szófaji egyértelműsítés. In Tanács, A. and Vincze, V., editors, *VII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2010)*, pages 275–283, Szeged.