A Maximum Entropy Approach to Proper Name Classification for Norwegian

Åsne Haaland
Dr Art thesis
Department of Linguistics and Scandinavian Studies
University of Oslo

March 13, 2008

Series of dissertations submitted to the
Faculty of Humanities,University of Oslo
No. 335

ISSN 0806-3222

Cover: Inger Sandved Anfinsen.
Printed in Norway: AiT e-dit AS, Oslo, 2008.

Produced in co-operation with Unipub AS.
The thesis is produced by Unipub AS merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

Unipub AS is owned by
The University Foundation for Student Life (SiO)

Acknowledgment

I would like to thank Professor Janne Bondi Johannessen, supervisor at my home institution, for recruiting me from German linguistics to computational linguistics. I would also like to thank her for creating a nice working atmosphere. Secondly, I would like to thank my supervisor in Sweden, Professor Joakim Nivre, for his very thorough and clear comments and suggestions. The very high standard of his own research, is something to aspire to.

When I started my PhD work, I was the first PhD student in computational linguistics in the department. This, together with the fact that I had never studied computational linguistics per se, posed extra challenges. I am heavily indebted to Anders Nøklestad who played a crucial role in my learning to program. I would also like to thank Lars Nygaard who helped me in learning to program in a more structured way and for letting me use a Perl module of his. Thanks are also due to Andra Björk Jónsdottir and Kristin Hagen for introducing me to Emacs and Unix and for annotation discussions. Finally, I wish to thank Zhaleh Feizollahi for proofreading this thesis.

My colleagues at the then Department of Linguistics made it a very nice place to be from day one. I would like to thank everybody who told me about their ongoing research, often in very different fields from mine, but inspiring none the less.

In closing, I would further like to thank my parents, Hilde and Arne, for their support.

This thesis work has been funded as part of the Language Technology Program of the Nordic Council of Ministers administered by NorFA (now NordForsk).

Oslo, September 25, 2007

Contents

Chapter 1

Introduction

Named Entity Recognition (NER) is the task of recognizing single words or multi-word expressions as instances of a predefined set of semantic categories, the named entities. Named Entities (NEs) comprise classical classes of proper names such as the names of people, organizations and locations which represent the most common name types in general news text. Numerical expressions (dates, currency, percentages) represent another group of NEs. But NEs can also be domain specific: in the biomedical domain, possible NEs are names of DNA, protein, cell type, etc. In example (1), NER means that *South Africa* would be recognized as a country name, *Leremi* as person.

(1) *South Africa*'s *Leremi* dies in car crash.

There is wide consensus in the language technology community that NE detection and classification may improve the performance of many language technology applications such as question-answering, information extraction, machine translation and topic detection and tracking. NER represents a mature field of research internationally.[1] A wide range of machine learning and rule-formalisms have been employed. There is no reason why NER should not also be performed on speech, but we will in this thesis let NER equal NER on text.

In example (1), *South Africa* could be recognized as a country name by consulting a list of country names, while the semantic category of *Leremi* is determined by its neighboring words 'dies in car crash'. It is well recognized that both name-internal and name-external features can provide clues to the semantic category of the NE. All existing NER systems therefore combine features of the NE and its context to correctly identify the NE. In addition

[1] A special 2007 issue of the journal Lingvisticae Investigationes is entirely devoted to NER illustrating the continued interest in NER.

to lists and left and right neighboring words, NER systems for languages such as English, Spanish and Dutch typically employ information such as spelling features of the NE. Suffix information of the NE and its neighbors is also commonly used. Co-reference information, that is information of the semantic category of a second instance of the same name, has also proved useful for NER. We refer to this information as attributes or features.

1.1 Approach

This thesis examines automatic semantic classification of proper names for Norwegian general text. The proper names are detected by a rule-based grammatical tagger prior to the classification task. The following six name categories constitute our named entities: PERSON, ORGANIZATION, LOCATION, EVENT, WORK and OTHER. The OTHER category applies to names which do not fit into any of the other five categories. No sub-categories are used.

The semantic classification is based on maximum entropy modeling. Maximum entropy modeling represents a machine learning technique that has already been employed for Named Entity Recognition, though no such results have been published for Norwegian. This study examines which attributes of the name and its context contribute to a correct classification.

We have NE-annotated a tagged Norwegian corpus of 7 500 proper names employing the six semantic name categories. The corpus of 230 000 tokens is made up by excerpts of contemporary fiction as well as articles taken from a wide range of printed media. The name categories PERSON, ORGANIZATION and LOCATION are frequent in the corpus, while there are few instances of the other three categories, WORK, EVENT and OTHER. The rule-based name finder does not necessarily fully disambiguate, hence a token can receive more than one reading, but readings are ranked. We have overruled the tagger, so that all, but only actual proper names, receive a name category, hence the detection of names is perfect.

The NER task is made more difficult by our choice of annotation strategy which weighs context over surface form: while most NER systems would tag *South Africa* in example (1) as location (geopolitical entity), we would tag it ORGANIZATION if *South Africa* actually stands for the national sports team, and reserve the LOCATION tag for cases such as (2). This clearly reduces the effectiveness of list look-up.

(2) Three million Zimbabweans are thought to have fled to *South Africa*.

An off-the-shelf implementation of maximum entropy modeling has been applied for training and testing. This software employs the traditional algorithms for weight estimation and smoothing, namely the GIS algorithm and feature selection using a frequency threshold. Results are recorded for ten-fold cross-validation and measure both overall and category-wise performance. Attribute selection was based on the default values of the software. First, the results for single attributes were recorded, before the most effective attribute in turn was combined with different attributes. Statistical testing is employed in order to avoid attribute redundancy. Finally, attributes were added incrementally until a full classifier was reached, which in turn was optimized in terms of software parameters.

1.2 Research Questions

Here, we shall have a brief look at the main research questions. Only attributes derived from the same sentence as the NE (the name) were examined. The grammatical tagger which detects the proper names provides lemmatization and grammatical category for each token and attributes are derived from both the unprocessed text and from the grammatical tags. Upon this background, the following questions will be asked:

- What is the effect of only providing the name, alternatively, of adding neighbors to the name?

- Is the best performance achieved with the same number or with different numbers of left and right neighbors?

- How are names and non-names best represented lexically, as inflected forms or as lemmas?

- Does the grammatical category of neighbors to the name matter to performance? Given that the grammatical tagger frequently provides more than one reading, should we choose the grammatical category of the top-ranked tag, or should we preserve the ambiguity and record the grammatical categories of more than one reading?

- What is the contribution of the suffix of the name and the neighbors? We examine results for different number of neighbors of suffixes of varying length.

- In Norwegian, by convention, only the first part of multi-word names that denote public institution is capitalized. All parts of multi-word names of people and companies are on the other hand capitalized. We test if the distribution of capitalization across multi-word names is an useful attribute.
- What is the effect of recording if names are acronyms?
- What effect does the addition of list look-up have?

We are always interested in the effect to the different semantic categories. In the remainder of this chapter, we outline the structure of this thesis.

1.3 Outline

Chapter 2 discusses named entities, maximum-entropy-based NER, attributes, and finally the Nordic umbrella project for NER.

Maximum entropy modeling is the topic of Chapter 3. This is a supervised, probabilistic machine learning method. The presentation focuses on the more orthodox version of the maximum entropy model as this is what we use in this thesis work, but we show that modifications of the model exist.

Chapter 4 discusses the implementation of the maximum entropy model and cross-validation. The input format (the attribute representation) required by the off-the-shelf maximum entropy software is described. The default parameters for training the model are specified, as are the prediction (output) options. We specify how cross-validation performance is measured. Given two sets of cross-validation results, we would like to know if the difference in results should be attributed to chance. The chapter includes a discussion of how this can be statistically tested.

The Norwegian name annotated corpus is discussed at length in Chapter 5. We give a detailed account of the POS-tagged corpus. We also describe the set of named entities and how each label is to apply. The second question not only involves the coverage of each label, but includes how metonymy is dealt with.

Chapter 6 discusses our experiments and is our primary results chapter. The larger part of the chapter discusses attribute selection. The cross-validation results for different attributes, using the default model-building parameters, are recorded. A description of the different attributes and how they are implemented is provided. Results are presented in three steps: first, results for single classes of attributes are recorded. These results vary greatly.

Second, the most important attribute is combined with an additional attribute, the second attribute alternating between each of the remaining attributes. This time around results vary considerably less. Finally, attributes are added one at a time until a full model is reached. In a final section, we examine if the performance of the full classifier improves if the parameters of the learning algorithm are optimized.

Chapter 7 examines the performance of the optimized classifier. Any classifier can be expected to perform better on NEs encountered in the training data than on unknown NEs. We record how big the difference is for our classifier. The overall cross-validation results are compared with the results of alternative proper name classifiers for Norwegian and for Danish and Swedish. This amounts to a comparison on highly similar languages with related tag sets.

In Chapter 8, which constitutes the final chapter, we sum up our findings.

Chapter 2

Named Entity Recognition

This chapter provides background for the thesis project. To start with, examples are provided for varying set sizes of the NEs employed and their degree of sub-division. An overview of the many machine learning techniques that have been employed for NER is provided. Some existing NER systems based on maximum entropy modeling are described, none of them for Norwegian. Attributes of the context and the NE that are commonly used for NER are described. Our thesis project is part of a larger project for NER in three highly similar languages. We outline this project and look at the status for NER in Norwegian and describe two existing systems, one based on machine learning, the other rule-based.

2.1 Named Entities

The Sixth Message Understanding Conference (MUC-6) in 1995 pioneered NER by defining the task.[1] The following seven Named Entities were defined: proper names denoting people, organizations and locations (the three most frequent types of proper names in news text), plus four kinds of numerical expressions (date, time, money and per cent). NER was for English and domain specific.

Some tasks motivate dividing categories into subcategories. If we, for example, were to use our Named Entity Recognizer as part of a system that decides if a person name occurring in several documents denotes the same person or not, it could be useful to split the person category into politician,

[1] The MUC-6 web site can be accessed at http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html, the web site of the follow-up conference MUC-7 at http://www.itl.nist.gov/iaui/894.02/related_projects/muc.

entertainer, athlete, businessman, etc. This information could serve to establish if the name is used to denote different people across documents or not. The underlying assumption is that a name that is judged to denote a politician in one document, while it denotes an artist in another document, probably denotes two different people. Some people naturally do not fit into this grid: examples might include actors or professional athletes turned politicians like *Arnold Schwarzenegger*. Fleischman and Hovy (2002) classify the person name category into eight subcategories: athlete, politician/government, clergy, businessman, entertainer/artist, lawyer, doctor/scientist, and police. For an example of cross-document person name disambiguation, see (Niu et al., 2004).

An example of domain specific NER can be found in the bio-medical domain, for example, Kazama (2004), who recognizes 23 different leaf entities.

Number-wise at the far end of the scale is Sekine and Nobata (2004) who recognize 200 leaf entities. Sekine and Nobata (2004) define a taxonomy, an Extended Named Entity Hierarchy, with the aim of recognizing a total of 200 leaf entities with three top-nodes. This enables them to offer on-demand information-extraction and question answering by allowing the client to choose among 200 possible entities. This amounts to multi-domain-specific NER for Japanese and English. With such a high number of NEs, supervised learning does not lend itself as a feasible alternative because of the annotation burden involved in providing instances for 200 different NE.

2.2 Machine Learning and Maximum Entropy NER

All machine learning presupposes a *training corpus* $\mathcal{T}$ of instances that are represented as attribute values, $\mathcal{T} = \{t_1, ...t_Q\}$, where each instance t_q is represented as $t_q = \{a_1...a_m\}$. The system is to learn to classify (predict the category of) new instances. Learning equals finding the weight w_i for each attribute a_i. The category is assigned based on the highest combined weights. In the case of some learning techniques, such as maximum entropy modeling, weights are combined to form a probability.

Learning is *supervised* if the learning instances come with the correct category. Supervised learning in general is more successful than unsupervised, but involves manual annotation.

A number of different machine learning techniques have successfully been applied to NER. Some examples are Support Vector Machines (Isozaki and

Kazawa, 2002), Decision Trees (Sekine et al., 1998), Hidden Markov Models (Bikel et al., 1997; Zhou and Su, 2002) and Memory Based Learning (Sang, 2002). Active learning represents a means of reducing the amount of annotation needed: automatically choosing batches of examples for annotation after certain criteria alternates with re-training. Shen et al. (2004) is an example of NER, active learning and Support Vector Machines. It is also possible to combine several learners in the form of boosting (Carreras et al., 2003) or ensemble learning (Kim et al., 2002).

The above presentation of machine learning techniques other than maximum entropy modeling is cut to the bone. However, we describe a memory-based system in detail in section 2.4, and also show some of the variation in Machine Learning and NER by elaborating on different versions of maximum entropy modeling.

2.2.1 Maximum Entropy-Based NER

Several existing named entity recognizers, but for languages other than Norwegian, are based on maximum entropy modeling.[2] Maximum entropy modeling was first used for Named Entity Recognition by Mikheev et al. (1998) and Borthwick et al. (1998). The two systems are both MUC-7 entries. Borthwick et al. (1998) stress that maximum entropy modeling lets one combine diverse knowledge sources without having to consider if features are independent in the probabilistic sense. Both do named entity recognition in one step, on raw text, and both are systems for English.

Mikheev et al. (1998) and Borthwick et al. (1998) combine a statistical method with hand-crafted rules. Characteristic of the Mikheev et al. (1998) system is the tight integration of rules and statistical analysis in the form of alternation between surefire rules and statistical analysis. The policy is to avoid making premature decisions by resolving ambiguity before a decision is made. Borthwick et al. (1998) let a maximum entropy NE-recognizer function as a post-processor to one or more rule-based named entity recognizer(s). The idea is that the maximum entropy component is to learn and hopefully correct the mistakes of the preceding rule-based system(s).

The next chapter is devoted to the maximum entropy model. Borthwick (1999) employs a traditional maximum entropy model. Malouf (2002) and Kazama (2004) examine alternative versions of maximum entropy modeling applied to NER. Interestingly, Kazama (2004) in the case of maximum

[2]Danish and Swedish strongly resemble Norwegian. No NER system based on maximum entropy exists for these two languages.

entropy-based POS-tagging, exploits un-annotated data in addition to the annotated. Kazama (2004) represents NER for the biomedical domain, some twenty NEs including protein, DNA, virus, body parts and tissue.

2.3 Attributes

All existing NER systems exploit a combination of features of the NEs themselves and features of the context. This corresponds to internal and external features in the terminology of McDonald (1996). The features used by a system may in reality overlap. We focus in this section on attributes used by NER systems based on machine learning. The majority of the systems also detect the NEs either as a separate stage preceding classification or in a combined operation. We here try to exclude attributes that seem only relevant to detection. The following features are more or less standard for machine learning based NER for a European language where person, organization and location names constitute the NEs.

While many NER systems use lists, the size and the quality of these lists have varied greatly. Mikheev et al. (1999) examine the effect of lists for English journalistic text with mixed cases. The named entities of this study are names of persons, organizations and locations. Results are reported for identical systems except for the kind of lists involved. (This system is described in section 2.2.)

Table 2.1: The result table is from Mikheev et al. (1999). While the full lists of 45 000 names yield the best results (leftmost column), providing some very common location names is dramatically better than no lists.

	Full lists		**Some locations**		**No lists**	
	recall	precision	recall	precision	recall	precision
organization	90	93	87	89	86	85
person	96	98	90	97	90	95
location	95	94	85	90	46	59

The best results, shown in the leftmost column of Table 2.1, are achieved for extensive lists, where the total number of names equals 45 000. On the other hand, if no lists are used, the results for the ORGANIZATION and PERSON categories are decent: recall and precision are 86 and 85 respectively for organization, 90 and 95 for person, while the corresponding figures

for location are only 46 and 59 (results in the rightmost column).[3] Only the categories PERSON and ORGANIZATION carry sufficient internal and external cues for a correct disambiguation. If, instead of no lists, a list of some 200 very common location names is employed, then the recognition of PERSON and ORGANIZATION improves somewhat, while the results for LOCATION improve dramatically: recall is up by 40, precision by 30 percentage points. Although common location names such as *France* contain few internal cues, they, since they are expected to be known to the wider audience, carry few contextual cues to name category.

A point not made by Mikheev et al. (1999), but which is argued in for example Johannessen et al. (2005), is that the usefulness of lists also depends on the mark-up strategy chosen, how surface form is weighted against context. Sports teams commonly carry location names. If, for example, names of sports teams are consistently counted as ORGANIZATION, this clearly makes a location name list less powerful.

External information includes a shifting lexical window anchored at the Named Entity with the two or three closest neighbors in either direction. Suffix information is standardly used both to detect and classify NEs. As for the internal features, the MUC-7 showed that orthographic features of the NE are useful for the recognition of person, location and organization names. Examples of such features are *acronym* which applies to acronyms such as *NATO*, *interjected-capitalized-letter* as in *NordForsk*, and the presence of non-letters or non-digits such as the ampersand or a period.[4]

We have so far described attributes that are derived from the same sentence as the NE. We term same-sentence information local, whereas different-sentence information is termed global. Co-reference resolution amounts to identifying a second instance of a name either in its full form or in a short version and can also include identifying the equivalent acronym. A second instance of a name can clearly occur outside of the sentence containing the first instance. Co-reference resolution is a crucial feature of the maximum entropy-based NER of Mikheev et al. (1998). Borthwick (1999) among others report that co-reference attributes improve performance. In the latter system co-reference resolution represents a separate post-processing step. Chieu and Ng (2002) demonstrate that global information, for example co-reference resolution, can be successfully incorporated into the same maximum entropy model which exploits local information.

[3]The measures recall and precision are defined on page 28.

[4]In Bikel et al. (1997) features are given different precedence, whereas in most systems many features may be active at once.

Table 2.2 shows attributes commonly used for NER.

Table 2.2: Standard Attributes

Name lists
Lexical Windows
Orthographic Features
Co-Reference

2.4 NER for Norwegian and the Mainland Scandinavian Languages

This thesis is part of a joint Nordic project for proper name recognition for the mainland Scandinavian languages, Danish, Swedish and Norwegian. The project, termed Nomen Nescio (NN), ran from 2001-04 and involved staff, PhD and master students from several Nordic universities.[5] Norwegian, Swedish and Danish are similar to the extent of being mutually intelligible. They constitute the national languages of Norway, Sweden and Denmark respectively. Of written Swedish and Danish, the latter resembles Norwegian most strongly. Until the launch of this network only smaller projects had been carried out on the mainland Scandinavian languages, and these were all for Swedish: Dalianis and Åström (2001) and Kokkinakis (2001). The stated aim of the project was to develop NER systems for a Nordic language in parallel, using different methods, but a shared set of NEs (Johannessen, 2004). The more theoretical aim of the project was to be able to infer something about the quality of each method, given the similarity of the languages involved and the shared NE set.[6] The project resulted in three name classifiers for Norwegian, two NE recognizing systems for Danish and one for Swedish. Statistical learning methods were limited to two systems for Norwegian. We

[5]The joint project home page is found at http://g3.spraakdata.gu.se/nn/.

[6]An alternative would have been to first develop an NE recognizer for one of the three, and then exploit this classifier to develop NE recognizers for the two remaining related languages. In Solorio (2005) a hand-coded NE-classifier for Spanish is enforced with machine learning to become an NE recognizer for the related language of Portuguese, while Carreras et al. (2003) develop NE finders for Catalan from a Spanish NE finder. In the latter case the lexical features of the Spanish system are translated to Catalan, or alternatively the NE finder is trained on a mixed corpus of Spanish and a dramatically smaller corpus of Catalan.

describe here the two Norwegian systems developed alongside ours, as well as provide a superficial description of one of the Danish systems.

As the project started, there existed a grammatical tagger for Norwegian, namely the Oslo-Bergen tagger. As part of the project the tagger's ability to detect proper names was enhanced. Names that are potentially difficult to detect are sentence-initial names, since non-names are also capitalized in this position, as well as multi-word names, as Norwegian allows non-first parts of the name to be non-capitalized. It was therefore natural to employ the grammatical tagger to detect the names, and to develop pure classifiers.

Two proper name classification systems for Norwegian exist in addition to ours: one based on machine learning, Nøklestad (2004), and one rule-based, Jónsdottir (2003).[7] Norwegian has two written standards, nynorsk and bokmål, with bokmål as the dominating standard. The two systems are, like our system, for bokmål. The three systems share a number of important characteristics. Firstly, all three systems are classifiers only, which presuppose that the names are detected by the same grammatical tagger. The tagger is rule-based and does not necessarily disambiguate completely, meaning that a token may receive more than one reading. Secondly, they are developed using the same annotated data. Still, it is the system based on supervised machine learning that compares most directly to ours. Unlike the rule-based, it assigns a unique category to the NE. Moreover, the attributes used resemble ours, and results are, as in our case, recorded for cross-validation.

The Nøklestad (2004) system for Norwegian proper name classification is memory-based. Memory-based learning, which also goes under the terms instance-based or analogy-based learning, represents non-probabilistic supervised learning. Training simply equals storing the instances. Categories are assigned based on a similarity metric and the k nearest neighbor(s). A common choice of k is one, ie a new instance is assigned the category of the most similar instance in the training data. Four different k-values (5, 11, 19 and 25) were each combined with each of four similarity metrics (information gain, gain ratio, chi-squared and shared variance). With a complete attribute set, a k-value of five gives the best results. The performance of Nøklestad (2004) is given in Chapter 7.

Jónsdottir (2003) employs the same technique as the name-detection module, section 5.3. Constraint grammar (CG) is characterized by not demanding

[7]There is also the more recent example of Røyneberg (2005). This master's thesis employs rules to detect location names in Norwegian text. This system also employs the Oslo-Bergen tagger.

a full disambiguation, so that names in the system output will have a varying number of categories. Constraint grammar rules either select (map) or discard (disambiguate) alternatives. Jónsdottir (2003) comprises rules for four of the six categories: PERSON, ORGANIZATION, LOCATION and WORK, where WORK denotes products of media/the arts. This means that rules involving the EVENT and OTHER categories are missing with the exception of the default rule that maps all six categories to a name.

If the lexicon employed includes semantic information, the CG-rules can be formulated to exploit this information directly: if, for example, the lexicon entry of a verb states that its object must be inanimate, a rule can be formulated that says a name in this position cannot represent a person. However, the lexicon employed by the Norwegian tagger that detects proper names, does not include semantic information. For this reason Jónsdottir (2003) establishes classes of, for example, verbs or nouns that are to behave similarly with respect to a rule. At the time there did not exist a full systematic semantic lexicon, but Jónsdottir (2003) was able to use existing parts of such a lexicon to assemble the semantic classes.[8]

The mapping rule that assigns all the six proper name categories to each name constitutes the default rule. Rules include heuristic rules, which are characterized as being more general, and potentially more dangerous, than the ordinary rules. Heuristic rules are therefore applied after the ordinary CG-rules. A total of 110 disambiguation rules and 27 mapping rules constitute the system. However, the system represents a prototype and more rules are needed. Since there are few rules, this leaves us with a high degree of ambiguity. Jónsdottir (2003) examines different schemes for solving ambiguity further in a post-processing step. For example, the counts of the categories of the fully disambiguated instances of a particular name are recorded, and the most common category is assigned to the ambiguous instances of this name.

It was envisioned that the CG-system could be combined with a statistical one: the statistical system could solve the ambiguities left by the rule-based system. This represents a set-up like Borthwick et al. (1998) where a rule-based NER system precedes a statistical NER system. For this reason high recall was seen as more essential than high precision.[9]

Like Jónsdottir (2003), one of the Danish systems employs CG-rules. The

[8]The Danish Simple Lexicon has been translated into Norwegian, see Fjeld (2001).

[9]Recall and precision are in this case not identical to the recall and precision used by us to report results of the maximum entropy-based classifier: in the first case, the two measures are calculated on the basis that the names are assigned a varied number of categories.

main difference between the two systems is that Bick (2004) is able to write rules that directly employ semantic information in the lexicon. Moreover, name lists employed are more extensive than the ones used by Jónsdottir (2003) (section 6.1.6), and are also given more power. For example, sports teams that carry location names are tagged as locations and not as organizations. Finally, the Danish system comprises a much larger number of rules than its Norwegian counterpart.

Chapter 3

Maximum Entropy Modeling

Maximum entropy modeling represents a supervised probabilistic learning technique. This chapter focuses on model building, that is how to find our conditional probability estimate given some annotated data. Prediction itself is straightforward: the predicted category equals the most probable category according to the model. We use an off-the-shelf implementation to build our classifier. This implementation is described in the next chapter.

We start by introducing the Maximum Entropy principle, which is a principle for choosing probability estimates in the presence of annotated data. We demonstrate why the weights of the features represent Lagrange multipliers. We explain how this principle translates into mathematics and discuss how a unique maximum entropy model can be computed: there is no closed form solution, so weights are found through iterative scaling. Several such algorithms exist, while our software employs the GIS algorithm. We explain why the maximum entropy model is susceptible to over-fitting, which means that the accuracy of the classifier is less than optimal as it too closely fits the training data, and discuss possible counter measures. Berger et al. (1996) provide a good introduction to the maximum entropy framework.

3.1 The Maximum Entropy Principle

We illustrate the Maximum Entropy Principle through an example. Assume that there are six semantic name categories: PERSON, ORGANIZATION, LOCATION, WORK, EVENT and OTHER. Assume also that a name-category annotated corpus comprises five instances of the name *Jordan*, and that three of the five instances are tagged as LOCATION, whereas the remaining two have been judged to denote a PERSON. Now our principle

says that the probabilities of the two categories LOCATION and PERSON given that the name (w0) equals *Jordan* should be set equal to:

(3) $p\,(x = \text{LOCATION} \,|\, \text{w0} = Jordan) = 0.60$
$p\,(x = \text{PERSON} \,|\, \text{w0} = Jordan) = 0.40$

An additional fact of the training data may be that there are ten instances for which an inflected form of the verb *visit* immediately precedes the name. We let *l-1* stand for the preceding lemma. The name in four cases carries the PERSON tag. The remaining six names were equally divided between the LOCATION and the ORGANIZATION categories, ie there are three instances of each. This fact of the training data demands of the estimate that

(4) $p\,(x = \text{PERSON} \,|\, \text{l-1} = visit) = 0.40$
$p\,(x = \text{LOCATION} \,|\, \text{l-1} = visit) = 0.30$
$p\,(x = \text{ORGANIZATION} \,|\, \text{l-1} = visit) = 0.30$

Moreover, the principle says that for cases for which there are no statistics in the training data the estimate should assign identical conditional probability to each of the name categories. Berger et al. (1996) stress how this principle is in accordance with common sense: we should incorporate what we know of relative frequency, but not pretend to know if either of the alternatives are more probable than the other by setting the probability of one higher than the other.

The Maximum Entropy Principle has two parts. First, it imposes constraints on our choice of estimate by stating in what points the estimate must equal the empirical distribution $\tilde{p}\,(a, x)$. Second, among the family of probability distributions that fulfill these requirement it says which to choose. We are to choose the probability distribution which equals our observations, but which also has *the most evenly* divided probability mass. A maximally evenly distributed probability mass, which may be described as a maximally *uniform* or *flat* distribution, equals a model with maximum entropy.[1]

Entropy is denoted by a H. The entropy of a conditional distribution $p\,(x|\,a)$ where x and a denote category and attribute respectively is given in

[1]In information theory, entropy measures uncertainty. How uncertainty relates to uniformity can intuitively be seen: in the case of a uniform distribution, all outcomes are equally likely, hence there is maximum uncertainty about the actual outcome. The less uniform the probability distribution is, the more strongly one or certain alternatives are favored, the better chance of prediction.

(3.1),

$$H(X|A) = -\sum_{a,x} p\,(x,a)\,\log_{\,2}\,p\,(a|\,x) \tag{3.1}$$

Formalized, the Maximum Entropy Principle says to choose the probability $p^{\star}$ that is among the probability distributions $p\,(\mathcal{C})$ that satisfy the set of constraints $\mathcal{C}$, ie $p \in p\,(\mathcal{C})$, but which at the same time maximizes entropy, $H(X|A)$.

$$\text{Choose } p^{\star} \text{ such that } p^{\star} = \underset{p \in p\,(\mathcal{C})}{\operatorname{argmax}}\, H(X|A) \tag{3.2}$$

3.2 Features and Constraints

Berger et al. (1996) very illustratively show that already with a small number of facts of the training data, the constraints appear contradictory, so that the probability cannot be found analytically. We need to express the constraints $\mathcal{C}$ and to find the model of equation (3.2). The manually annotated data represents knowledge of how a category depends on some factor, for example how a name category depends on the immediate neighboring words of the name. In maximum entropy modeling, features are used to express the particular combination of a name category and a characteristic of the name or its surroundings found in an instance of the training data.

A feature is a binary indicator function, a function that takes two arguments, namely attribute value a and category x, and reserves the value 1 for a particular combination of the two.

$$f_i(a,x) = \begin{cases} 1 & \text{if } a = \text{attribute value is true and x = category} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

If we return to our example situation, we can assume that the sentence *The prime minister visited China.* represents one of the instances where an inflected form of *visit* precedes the name. The proper name *China* has been manually tagged as LOCATION. The feature $f(\text{l-1} = \text{visit}, \text{LOCATION})$ equals 1, in this particular case.

Each (selected) feature imposes a constraint: the expectation of the feature relative to the estimate p must equal its expectation relative to the empirical distribution $\tilde{p}$. $E_p(f_i)$ denotes the expectation relative to the model, while $E_{\tilde{p}}(f_i)$ denotes the expectation relative to the empirical distribution.[2].

$$E_p(f_i) = E_{\tilde{p}}(f_i) \tag{3.4}$$

[2]We will clarify what is meant by *selected* feature in section 3.5

where the left hand side equals

$$E_p(f_i) = \sum_{a,x} f_i(a,x)\, p\,(x,a) \approx \sum_{a,x} f_i(a,x)\tilde{p}\,(a)\, p\,(x|a) \tag{3.5}$$

The expectation of the same feature relative to the empirical distribution is

$$E_{\tilde{p}}(f_i) = \sum_{a,x} f_i(a,x)\, \tilde{p}\,(x,a) \tag{3.6}$$

3.3 Constrained Optimization

The task is to find the probability distribution of equation (3.2) that maximizes entropy, but which also satisfies the constraints imposed by the data. This equals a task of constrained optimization. The standard way of solving problems of constrained optimization is the method of Lagrange multipliers, which explains why the weights of the features in the maximum entropy model are Lagrange multipliers λ_i. Next follows a demonstration of the Lagrange machinery, so it is possible to go straight to equation (3.10), which gives the family of distributions which satisfy the constraints.

The strategy of the Lagrange Method is to solve the original equation (3.2), by maximizing a different equation and then substituting the answers back into the original equation. The first equation, here (3.2) is standardly referred to as the *objective* or *primal* function, whereas the second function, which we maximize, is referred to as the *dual* function (3.11). In many cases of constrained optimization, the dual equation is easy to solve, but, in our case, it can only be solved using numerical methods.

The first step of the Lagrange method is to construct the Lagrangian $\Lambda\,(p, \lambda_1, ...\lambda_F)$, which equals $H(X|A)$ plus the sum of each constraint (expressed so that it equals zero) multiplied by a Lagrangian multiplier λ_i.

$$\Lambda\,(p, \lambda_1, ...\lambda_F) = H(X|A) + \sum_i \lambda_i\,(E_p(f_i) - E_{\tilde{p}}(f_i)) \tag{3.7}$$

We now compute the unconstrained maximum of the Lagrangian relative to p and relative to the Lagrangian multipliers.[3] Holding the $(\lambda_1, ...\lambda_F)$ fixed, we compute the unconstrained maximum of the Lagrangian $\Lambda\,(p, \lambda_1, ...\lambda_F)$ over all p. We denote by $p_{\tilde{\lambda}}$ the p where $\Lambda(p, \lambda_1, ...\lambda_F)$ reaches its maximum, and by Ψ the corresponding value of the Lagrangian. $\Psi\,(\lambda_1, ...\lambda_F)$ is the dual function.

[3]Find partial derivatives by successively holding p and the different multipliers fixed and solve for partial derivative equals zero.

$$p_{\tilde{\lambda}} = \underset{p}{argmax}\, \Lambda\left(p, \lambda_1, ...\lambda_F\right) \tag{3.8}$$

$$\Psi(\lambda_1, ...\lambda_F) = \Lambda\left(p_{\tilde{\lambda}}, \lambda_1, ...\lambda_F\right) \tag{3.9}$$

Now calculus gives

$$p_{\tilde{\lambda}}\left(x|a\right) = \frac{1}{\sum\limits_{a} e^{\sum\limits_{i}\lambda_i f_i(a,x)}}\, e^{\sum\limits_{i}\lambda_i f_i(a,x)} \tag{3.10}$$

$$\Psi\left(\lambda_1, ...\lambda_F\right) = -\sum_{a}\tilde{p}\left(a\right)\, \log Z_\lambda(a) + \sum_{i}\lambda_i\, E_{\tilde{p}}(f_i) \tag{3.11}$$

The denominator of equation (3.10) is a normalization constant, which means that it ensures that the total probability is 1. It is identical to the $Z_\lambda(a)$ of equation (3.11). We see that only features, whose value equals 1, contribute to the probability of equation (3.10).

The Lagrange method guarantees that we find the maximum entropy model by solving the dual function (3.11). The dual function is smooth and concave, since it is the sum of two smooth and concave functions and this guarantees a unique maximum. A number of different optimization techniques may be used to find the Lagrangian multipliers λ_i.

3.4 Maximum Entropy Parameter Estimation

The classical algorithm for estimating the weights of the maximum entropy model is the Generalized Iterative Scaling (GIS) algorithm. It was initially introduced by Darroch and Ratcliff (1972).

The GIS algorithm requires that, for all pairs of attributes and categories, the features all add to a constant, C, equation (3.12). Now this is most often not the case, hence the need for a correction feature, that does not only take 0 and 1 as values, equation (3.14). In practice, $C^\star$ is maximized over the (a, x) pairs of the training data, although in theory $C^\star$ can be any constant greater or equal to the right hand side of equation (3.13). However, since $\frac{1}{C^\star}$ determines the rate of convergence of the algorithm, it is preferable to keep $C^\star$ as small as possible. F denotes the number of features.

$$\forall a, x \sum_{i} f_i(a, x) = C \tag{3.12}$$

$$C^\star = \max_{a,x}\sum_{i} f_i(a, x) \tag{3.13}$$

$$f_{F+1}(a,x) = C^{\star} - \sum_i f_i(a,x) \tag{3.14}$$

The algorithm is as follows:

1. Set ${\lambda_i}^{(0)}$ equal to an arbitrary value, say ${\lambda_i}^{(0)} = 0$ values which define the initial probability estimate.

2. Repeat until convergence:

$${\lambda_i}^{(t+1)} = {\lambda_i}^{(t)} + \frac{1}{C^{\star}} log \frac{E_{\tilde{p}} f_i}{E_{p^{(t)}} f_i} \text{ where (t) is the iteration index.} \tag{3.15}$$

Proof of monotonicity and convergence of the algorithm, which guarantees the existence of a unique maximum entropy model, is not included in this presentation, but see Berger et al. (1996) and references therein. GIS is the parameter estimation algorithm employed to build our classifier.

An alternative algorithm to GIS, which like GIS is especially made for finding the maximum entropy model, is the Improved Iterative Scaling (IIS) algorithm. The Improved Iterative Scaling algorithm tends to converge faster that the GIS, as it, unlike the GIS, does not require that features sum to a constant (equation (3.12)).[4] There is therefore no addition of correction features (equation 3.14). For the algorithm itself, see for example Berger et al. (1996).

While GIS and IIS are particular to maximum entropy modeling, it is also possible to compute the weights using *general* optimization techniques, a point made by Malouf (2002). Malouf (2002) advocates the use of the Limited Memory Variable Metric (LMVM) algorithm over GIS or IIS. Malouf (2002) compares parameter estimation algorithms on the basis of four different tasks. The algorithms are similar in terms of the accuracy achieved on the test data, but in terms of the numbers of iterations and training time the classical GIS and IIS are second to LMVM. Malouf (2002) states that better parameter estimation techniques can open up for sophisticated feature selection techniques.

3.5 Feature Selection

Over-fitting, which means that performance is less than optimal because the model too closely resembles the training data, is potentially a problem for all

[4] Curran and Clark (2003) show both analytically and numerically, that the correction feature, assumed to be required for the GIS, is actually unnecessary.

machine learning. We may therefore choose to omit parts of the data. The easiest approach is to omit low-frequency features on the assumption that they are unreliable or uninformative.

Daelemans et al. (1999) argue that rare events in the corpus can represent real sub-regularities, Berger et al. (1996) propose a more sophisticated form of feature selection than simple threshold cut-off. Features are added one by one, starting with the empty set. Each time a feature is to be added, all candidate features are evaluated in the following way: for each candidate the maximum entropy model is computed that corresponds to the already selected features plus the candidate feature. Then, the increase of log likelihood of the training data which the addition of this feature to the feature set represents is calculated.[5] The feature which corresponds to the largest increase in log likelihood is chosen. This procedure is repeated until a chosen stop criterion holds.

In order to reduce the computational load involved, Berger et al. (1996) propose to adopt the assumption that only the weight of the latest added feature must be computed, while the weights of the earlier added features are unchanged by the most recent addition. They term this procedure the Random Field approach. The Random Field approach estimates good estimates relatively fast. It does, however, not guarantee that we at every point add the best feature, because contrary to the underlying assumption as we add a new feature to the model, all parameters can change.

3.5.1 Smoothing with a Prior

It can be shown that the maximum entropy model is also a maximum likelihood estimate (MLE), see Berger et al. (1996) for a mathematical proof. This means that the maximum entropy model is the probability distribution of the family of exponential distributions of equation (3.10) for which the training data is maximally likely. Maximum likelihood estimates are attractive estimates, as they are consistent (in mathematical terminology).[6] That the maximum entropy model represents a maximum likelihood estimate further legitimates this choice of estimate.

Under the maximum likelihood estimate (MLE), the probability of any seen event is set high, while the unseen events (which are not in the training data) receive a probability equal to zero. This poses a problem when

[5]Log likelihood often replaces likelihood. An increase in log likelihood represents an increase in likelihood.

[6]For more on the MLE, see for example Rice (1995).

the MLE is used as an estimate in Natural Language Processing, as some words are very frequent, while the vast majority are very uncommon.[7] The solution to this problem is to transfer probability mass from the seen to the unseen events, by lowering the probability of the seen events. This transfer of probability is termed *smoothing*.

In the case of the maximum entropy model MLE, one way to lower the probabilities of the events of the training data is to use a prior. The maximum likelihood estimate is part of the frequentist tradition of statistics, whereas there also exists an alternative tradition: Bayesian statistics. The main difference between the two schools is that while the frequentists base the estimate entirely on the data, the Bayesian school incorporates a prior belief.

Chen and Rosenfeld (1999) propose to use the normal distribution (the Gaussian) with zero mean and equal variance ${\sigma_i}^2$ for all weights as a prior on the weights.[8] This smoothing method is termed Gaussian Maximum A Posteriori (MAP). Without a prior on the weights, the parameters that maximize the likelihood *lik* of the training data are chosen according to equation (3.16).[9]

$$\underset{\lambda}{argmax}\ lik(\lambda) = \underset{\lambda}{argmax} \prod_i P_\Lambda(x_i|a_i) \tag{3.16}$$

With a Gaussian prior, we instead find:

$$\underset{\lambda}{argmax} \prod_i P_\Lambda(x_i|a_i) \times \prod_i \frac{1}{\sqrt{2\pi{\sigma_i}^2}}\, e^{-\frac{{\lambda_i}^2}{2{\sigma_i}^2}} \tag{3.17}$$

where the second factor equals the Gaussian prior. In equation (3.17), the probability of the training data is ignored assuming it does not depend on the weights. Not ignoring this probability means dividing the above expression with this probability.

A modification of the maximum entropy model is clearly only viable if modified constraints can be expressed and it is possible to come up with an algorithm that finds the weights of the new model. Gaussian MAP relaxes the equality constraint as follows:

$$E_{\tilde{p}}(f_i) - E_p(f_i) = \frac{\lambda_i}{{\sigma_i}^2} \tag{3.18}$$

[7] For the problem of the MLE and the data sparseness in NLP, see Manning and Schütze (2000) pages 197-199.

[8] The normal distribution is the probability distribution which is known to the public: it is bell-shaped and symmetric, variance determines height and width.

[9] The underlying assumption is that the X_i are i.i.d. (independently and identically distributed). Their joint density is therefore the product of the marginal distributions (Rice, 1995, page 254).

where λ_i is the Lagrangian multiplier, ${\sigma_i}^2$ the variance of the Gaussian prior. A modified GIS algorithm can be employed to find the weights, see, for example Curran and Clark (2003).

The Gaussian prior has the effect of drawing the weights closer to zero than they would be without a prior. The probability of each instance of the training data is therefore lower than without the use of prior. Gaussian MAP does not, however, have the ability of feature selection, as it does not force weights to equal zero.

The Gaussian prior for maximum entropy models has been taken on or tested by other researchers, for example, Gaustad (2004). Curran and Clark (2003) show that using Gaussian MAP improves performance over a simple frequency threshold.

As for the choice of prior, Goodman (2004) argues in favor of replacing the Gaussian with the exponential distribution: the weights of the most frequently seen events are plotted. The shape of the plotted weights resembles the exponential distribution and not the Gaussian.

There are additional methods for countering over-fitting. For a discussion of the different methods, see for example Kazama (2004). Frequency threshold, Gaussian MAP, inequality constraints and combinations thereof are applied to text categorization and NER for English medical domain.

Our maximum entropy implementation exclusively employs Generalized Iterative Scaling for the estimation of the feature weights and frequency threshold cut-off for feature selection.

Chapter 4

Method

With this chapter we start on a new part of the thesis. In the next two chapters we will discuss methodological issues. While the subsequent chapter is devoted to issues concerning the annotated data for Norwegian, this chapter discusses two themes. The first subject constitutes the off-the-shelf maximum entropy software we use. The default model-building parameters are given, as are the alternatives for output, ie prediction. The format of the feature representation is also explained. The second subject of this chapter is less compact, but is made up by several related points. To start, the reasoning behind the re-sampling of the annotated data is explained, followed by how the re-sampling is done and a discussion of the result measurements. The purpose of this thesis is to investigate what features of the proper name and the context are most useful for an automatic classification. While we report test results for different classifiers on a given sample, we would like to be able to say which classifier can be expected to do best on any given sample. Additionally, we would also like to estimate how good a classifier is. We will therefore discuss what the possibilities are for model selection, which is what the comparison task amounts to, and for model assessment, given our use of re-sampling. We start this chapter by describing the software.

4.1 The Maxent Package

For training and testing we used the Maxent package, version 2.0, of the OpenNLP framework.[1] This is an open-source Java implementation of conditional maximum entropy modeling. We first describe the Maxent package relative to training, then to testing. The preceding chapter on maximum

[1]The Maxent web site can be accessed at http://maxent.sourceforge.net/.

entropy modeling showed that there is more than one way to estimate the weights and, if one wants instance editing, there are different options. Off-the-shelf software tends to represent more orthodox approaches: the Maxent package offers only the option of the global threshold for feature selection and the Generalized Iterative Scaling algorithm to estimate parameters. Three is the default threshold value: features must be seen at least three times in the training corpus in order to be included in the model. The default value for the number of iterations of the scaling algorithm is 100. As explained in Chapter 3 there are more recent alternatives for parameter estimation and feature selection, ie model building, than the Generalized Iterative Scaling algorithm and feature frequency cutoff. As for the lack of more recent alternatives, the focus of this thesis is on the relative importance of the attributes and not on techniques for parameter estimation and smoothing for maximum entropy models. It has been argued that the technology itself is not all that important, but that above all good features make for a good system.[2] For prediction, assigning a category to the name, the Maxent package offers two alternatives: it either outputs the most probable category, or all categories in fixed order with their respective probabilities.

4.1.1 Attribute Representation

Maximum entropy modeling relies on features, as seen in Chapter 3. Features are indicator functions that take attribute-value and name-category as arguments, and reserve the value 1 for particular argument combinations. The value of the feature is otherwise 0. The features are, however, not represented as 0s and 1s. The features of a name as input to this package are represented in the following way: it is only the features whose value is 1, that are represented. The attribute values are blank-separated, while the name's category is the rightmost element of the line representing the features of a particular name. This will become clearer with the following example:

(5) Han nevner Tyrkia som eksempel.
he mentions Turkey as example.
He gives Turkey as an example.

In the case of (5) we want to encode the following information: *Tyrkia* is the name in question, the two previous words are *han nevner* and the two

[2]This view is voiced by for example Christopher D. Manning (invited speaker to the CoNLL2004 conference) in his lecture titled Language Learning: Beyond Thunderdome. The lecture is found at http://www.cnts.ua.ac.be/conll2004/pdf/13838man.pdf

following are *som eksempel*. The category of *Turkey* is location. This is what the input to the package looks like:

(6) w-2=han w-1=nevner **w0=Tyrkia** w1=som w2=eksempel LOCATION

Attribute representation is discussed at length in Chapter 6. The leftmost element w-2=han illustrates how sentence-initial words that are not names are converted to lower case, so that a sentence initial neighbor does not differ from a non-initial neighbor in terms of capitalization. Additionally, we notice that the attribute values are identified: we write *w0=Tyrkia* instead of just *Tyrkia*, where w0 stands for the name itself. The reason is that the number of attribute values for each name varies. If we try to encode the same information for the following sentence, we get a different number of attribute values than above. The reason here is that we do not cross sentence-boundaries.[3]

(7) Men nå ser det ut til å svikte, mener Jørgensen.
but now appears it to fail, finds Jørgensen.
But it now appears to fail, finds Jørgensen.

(8) w-2=, w-1=mener **w0=Jørgensen** w1=. PERSON

Only in sentence (5) is the name sufficiently in the middle of the sentence to have two neighbors in both directions. Example sentence (7) lacks a neighbor to the right as the name is too close to the end of the sentence. It is not enough to only state their value. Neighbors may equal a clause boundary marker such as w1=. .

The final point we want to make here about the input format is that input for training and testing appears the same. The name category is the last element of the line in both cases. In the case of testing, the name category is ignored during prediction, but is then held against the predicted category by the classifier to evaluate its performance.

4.2 Re-sampling: Cross-validation

In case large amounts of annotated data are available, one can partition the data into three: the first part would be reserved for training, the second part for tuning the system, while the held-out data would serve for evaluation and

[3]Our script uses a Perl module by Lars Nygaard, University of Oslo, which builds the data structure one-sentence-at-a-time.

comparison of systems. The Norwegian name-category annotated corpus is the topic of Chapter 6. Relevant for us here is that the name-category annotated data for Norwegian amounts to 230 000 tokens of which some 7 500 are names.

Because of the limited size of the annotated data we use a kind of re-sampling. We therefore train and test using ten-fold cross-validation: we train and test on ten different 9 : 1 partitions of the corpus. The ten percent of the data that is used for testing in each run is unique to that run, while the ninety percent used for training partly overlaps with the training data of any other run. In this way all parts of the annotated data are used for both training and testing. A lot of training data is necessary if we are to report with some level of confidence a classifier's ability to learn. At the same time, a lot of test data is necessary are we to report results with a high degree of certainty. We made ten partitions respecting document boundaries. For more on the documents, see section 6.2.

Table 4.1: The number of names in the training and test data for the different folds. The number of names in the test data of each fold varies from 497 to 1052.

Fold	**Training data**	**Test data**
1	6837	695
2	6864	668
3	6616	916
4	6730	802
5	6480	1052
6	7035	497
7	6582	950
8	6878	654
9	6932	600
10	6834	698

Table 4.1 shows the number of names contained in the training and test data of each fold. The number of names in the test data, which equal the entries of the rightmost column, sum to 100 percent of the names in the annotated data, that is 7532. Each of the six name categories is represented in both training and test data of each fold. Ten percent of all the names equals 753 (754 in the case of two folds) but, as we said, partitions were made on the basis of entire documents, so the respective number of names

in the test data of folds can be considerably higher, for example, there are 1052 in fold 5, but only 497 in fold 6.

4.3 Performance Measures

The standard way of using maximum entropy modeling for prediction is to assign the most probable category to the instances of new text. Accuracy is measured by comparing the most probable category according to the model to the correct category. Accuracy is reported in terms of recall, precision and F-measure, whose definitions are given below. Cross-validation results are reported for each name category and for names overall. We do so as attributes might have different effects on different categories. In addition, as we will see in Chapter 6, some name categories are vastly more common than others. The most common category represents close to every second name, while the least common category is represented with only 39 names in the entire corpus. The overall score to a large extent reflects the results for the most common categories. For an actual example of how ten-fold cross-validation is reported, see Table 6.7 on page 74. Cross-validation results are given as the mean and standard deviation of the ten runs for each of the measures of accuracy. We follow standard notation and report results as the mean $\bar{X}$ followed by the standard deviation s in parenthesis. The definition of standard deviation is given here to demonstrate why a small sample, a small m, corresponds to a large standard deviation. Standard deviation approaches zero as m approaches infinity.

Definition 4.1 (Standard deviation)

$$s = \sqrt{\frac{1}{m-1}\sum_{i=1,m}(X_i - \bar{X})^2}$$

The square of the standard deviation s^2 equals the sum of the squared difference of an observation X_i and the mean $\bar{X}$ divided by sample size m minus 1. Standard deviation is often given in terms of the variance, which equals s^2.

Now for the three measures of accuracy: the recall for a certain category is the fraction of names of this category in the annotated test corpus, which was correctly marked up by the system. Its definition is given in 4.2.

Definition 4.2 (Recall)

$$R = \frac{|\textit{correct instances found}|}{|\textit{all instances in test}|}$$

If there are, for example, 50 instances of a certain category in the test data, and 40 of these are correctly identified by the classifier, recall equals $\frac{40}{50}(\times 100) = 80$. All accuracy measures are reported as percentage points. The precision of a category is the fraction of names marked up by the system to belong to that one category, for which the assignment is correct. The definition of precision is given in 4.3.

Definition 4.3 (Precision)

$$P = \frac{|\textit{correct instances found}|}{|\textit{all predicted instances}|}$$

If the classifier assigned a particular tag 60 times of which 40 were correct, precision equals $\frac{40}{60}(\times 100) = 67$. We want to report both recall and precision, since there is a trade-off between the two. If we assigned the same category to all instances to be classified in the test data, recall would be 100 percent for this one category, while precision would be low. If we on the other hand correctly assigned a small number of a certain category, precision would be high and recall low, assuming that many instances went undetected. The F-measure is a combined measure of precision and recall and is defined in 4.4.

Definition 4.4 (F-measure)

$$F_\alpha = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

where R denotes recall and P precision.[4]

We will weight recall and precision equally. With an equal weighting of P and R, i.e $\alpha = 0.5$, the F-measure simplifies to $\frac{2PR}{R+P}$.[5]

Oracle Accuracy

We saw in a previous section that the Maxent package outputs either the most probable category or a list of the categories with their respective probability. We reserved the term accuracy for when the most probable category according to the system is held against the correct category. In addition, we want to measure how often the correct category occurs among the n most

[4]In this definition P and R both vary between 0 and 1, while the range of P and R has so far been defined as [0,100].

[5]F_β is an equivalent measure. $F_\alpha = F_\beta$ in the case of $\beta = 1$ and $\alpha = 0.5$.

probable categories, where n varies from 1, the most probable category, to 6, the total number of categories. This we term oracle accuracy.

As for oracle accuracy we do not distinguish between ranks, ie we do not rate higher the case in which the correct category is found at the top of the list, rather than at the bottom. Neither do we distinguish the case for which two categories are equally probable, from when one category is more probable than the other. As with accuracy, we give results in terms of mean and standard deviation of the ten folds, but here it only makes sense to talk of recall, ie the fraction of names that were included in the n most probable categories for this instance. Focus is on reporting results in terms of accuracy. Oracle accuracy also shows if some categories generally are deemed more probable than others.

4.4 Generalizing Results: Statistical Analysis

We want to optimize a classifier in two stages: in the first stage the default values of the maximum entropy implementation are held fixed, while cross-validation results are recorded for different kinds of features. In the second stage the best features from stage one are kept, while the model-building parameters are optimized. We measure directly how our classifier performs on our data, but we are interested in a classifier relative to new data. Which classifier can be expected to perform the best on new data and how well do we expect the best performing system to actually be? As a reminder, we have no held-out data. Hastie et al. (2001) devote a chapter to model assessment and selection.

So what do we immediately know? It is reasonable to expect that highest scoring alternatives for our data to perform less well on an independent sample as parameters are tuned to the cross-validation data. As we will see in the results section of the next chapter, the standard deviations appear to be rather large. Hence it is not immediately clear how big the pairwise difference in the means must be in order to safely say that one system is probably better than the other. Many studies do not include any statistical analysis, but the function of statistical testing is to judge whether or not pairwise differences should be ascribed to chance or as alternatively representing real differences.[6]

[6]The principle behind all hypothesis testing is as follows: the probability of seeing the test statistic or a more extreme value given the null-hypothesis is recorded in the p-value. P-values that are smaller than the chosen level of significance level entail rejection of the

We will use McNemar's test in order to test if system A can be expected to generally do better than system B. Our choice of statistical test is the result of a two-part argumentation. First, because we are comparing two results obtained from testing on the same data, we need a paired test. The experimental set-up is paired as we twice employed the entire annotated corpus in the course of the ten runs. Second, cross-validation gives an artificially low standard deviation: the value of the standard deviation can because of the overlapping training data be expected to be smaller than in the case of ten independent runs. The second requirement to the test of choice points us to a non-parametric test. Justifications for our choice of McNemar's test are found in Dietterich (1998), which examines different paired tests. It is customary in the language technology community to apply McNemar's to cross-validation results.

McNemar's test is a non-parametric test, which means that it does not make any assumptions, assumptions that may not hold, as to how the cross-validation results are distributed. For example, it does not assume that cross-validation results are normally distributed.

McNemar's test is a 2 x 2 test, see Table 4.2. The following numbers constitute the four cells of the table: the number of times both systems made the correct classification n_{11}, the number of cases both systems got it wrong n_{22}, the number of instances system A got it right, while system B didn't n_{12} and finally the number of times system B and not system A was right, n_{21}. McNemar's test represents a chi-square test with 1 degree of freedom. The test statistic, ${\chi^2}_1$, equals $\frac{(n_{12}-n_{21})^2}{n_{12}+n_{21}}$. The null-hypothesis is that the probability of system A assigning correct category is the same as for system B. For further mathematical properties of the McNemar's test see Rice (1995).

Table 4.2: This table shows the input to McNemar's test. Only the counts n_{12} and n_{21} contribute to the test statistic.

	# system B correct	**# system B incorrect**
# system A correct	n_{11}	$\mathbf{n}_{12}$
# system A incorrect	$\mathbf{n}_{21}$	n_{22}

When statistically testing if one system is better than another, we instead

null-hypothesis.

of using the mean values as given by the cross-validation results, use the two numbers equalling the number of times one system assigned the correct category, while the other system got it wrong and when system two was right while system one was wrong. The numbers n_{11} and n_{22} are as we have seen, not part of the test statistic. This corresponds to common sense: it seems reasonable that the number of instances where both systems agree is not relevant for the comparison.

In the case of cross-validation, the test file of each fold is appended, and McNemar's test is performed on the resulting file. The maximum entropy model naturally varies somewhat from fold to fold as it is trained on somewhat different training data. McNemar's represents a Chi-square test. Chi-square tests demand that the counts of the cells are not too small, or the approximation becomes invalid. A rule of thumb says that the expectation value of each cell should be greater than five. As we are comparing the predictions of two different systems for a total of 7500 names, small cell counts should not be a problem. We further use two-sided McNemar's with Yates correction. Two-sided means that the alternative hypothesis takes the form of $H_A : p(A) \neq p(B)$. Yates correction is a way to circumvent the problem of too low counts for the Chi-square test, but may also be used in cases where counts are sufficiently high. We use a significance level of 0.01. It is of course only possible to compare results for different classifiers on a shared set of names. The test may not be applied to disjoint sets. We may use McNemar's to compare results for for example single part-names, but results for single-part names may for example not be tested against multi-part names using McNemar's.

To summarize, cross-validation is well suited for model selection since we can use the McNemar's test, while for model assessment we only know that the true value might be lower than the cross-validation results.

There are three methodological issues and we have in this chapter treated two of them. The first issue concerns the learner. The second concerns how performance on the cross-validation data is measured, and how general performance of different classifiers can be inferred from these figures. The cross-validation data constitutes the third issue and is topic of the next chapter.

Chapter 5

The Norwegian NE-annotated Corpus

This chapter concentrates on different aspects of the cross-validation data where each name carries a name category tag. To start, some background information on names in Norwegian with focus on capitalization is provided. No name-category annotated data existed for Norwegian so we made our own as part of the Nomen Nescio project (section 2.4). The work proceeded in four steps: first a corpus was established. Secondly, this corpus was tagged with a grammatical tagger for Norwegian. We then corrected the tagger output so that all names, but only names, are identified as proper names. The last step was to assign the correct category to each name. To a large extent this chapter follows these four steps. First, the texts that constitute the corpus are described. We then describe the POS-tagging of this corpus and illustrate what the tagger output looks like. We motivate the decision to correct the tagger output, and how the output is modified. We then sidestep to describe the strategies used by the grammatical tagger in order to detect proper names. A description follows of how we arrived at the set of name categories and which ones they are. We finally show which name types belong to a category and explain how metonymy is dealt with.

5.1 Norwegian Names: Capitalization or not

The general rule for Norwegian names is that they, unlike common nouns, are capitalized. A name distinguishes itself from a common noun in that a name refers to a specific object and gives its name, whereas a common noun refers to an object and tells what kind of object this is. In this section

we show that semantic expressions that are capitalized in English may have uncapitalized Norwegian counterparts. The second point we will be making is that non-first parts of Norwegian multi-part names are not necessarily capitalized. Then we follow with some observations on actual practice.

As we summarize the rules for uppercase-lowercase in Norwegian names we lean on Vinje (2004), which is an authorized normative description of Norwegian spelling.[1] Table 5.1 is a non-exhaustive list of types of names that must be capitalized. We see that names denoting people, locations, institutions and organizations are capitalized.[2] However, Norwegian has less absolute capitalization of names than English. Table 5.2 gives name types that in most cases are not capitalized. Such uncapitalized examples are terms denoting months, days of the week, ethnic groups, languages, members of organizations, treaties, laws in the natural sciences, etc.

Some name types have members in both categories, ie non-capitalized and capitalized members. There are three factors that give such a division. In the first case the uncapitalized word takes on a slightly different meaning than the capitalized counterpart. One such group is where the capitalized term denotes the institution, whereas the uncapitalized term refers to the person in office. To this group belong for example *Kongen* vs *kongen (the king)*, *Fylkesmannen* vs *fylkesmannen (the county governor)*, *Sysselmannen (the governor)* and *Barneombudet (the ombudsman for children)*. The second factor is the presence of a proper name as the first part of the name. When a proper name makes up the first part, otherwise non-capitalized name types are capitalized. In Table 5.2 for example illnesses, historical events and treaties are listed, in general, as lowercase, but in the case where a proper name constitutes the first part of the name, as in *Alzheimers sykdom (Alzheimer's Disease)* or *Versaillefreden (The Versaille Peace Treaty)*, the resulting expression is capitalized. Finally, the recentness of a loan-word or expression and the degree to which it has become part of standard vocabulary affects capitalization: in our corpus the religions *Santeria* and *Art of Living* were capitalized, while the more familiar *katolisisme (Catholisism)* is not. Similarly *Halloween* and the name of the Mexican *De dødes dag* (*The Day of the Dead*) were capitalized in our corpus. Once a term has been incorporated into Norwegian the initial capitalization may disappear as in: *halloween*, *aids*, *streptokokk*.

We have so far discussed initial capitalization. In the case of multi-word

[1] Vinje (2004) is authorized by the Council of Norwegian Language.

[2] Administrative units or sections within an institution or company should not be capitalized.

Table 5.1: Examples of name types that are **consistently** capitalized. They are therefore to be assigned category by our system.

Object type	**Example**
Person	Bill Clinton, Harry Potter
Individual animals	Flipper
Religious and holy personage	Frelseren (The Saviour)
Institutions, organizations, political parties, companies	Den norske opera (The Norwegian Opera), Care, Det norske arbeiderparti (The Norwegian Labour Party), Microsoft
Buildings, monuments	Eiffeltårnet (The Eiffel Tower)
Celestial bodies, stars	Venus
Geographical locations	Norge (Norway)
Books, magazines	Newsweek
Religious texts	Skriften (The Scripture), Toraen (The Torah)
Flags	Trikoloren (The Tricolor)
Boats, spacecraft	Endeavour, Challenger

Table 5.2: Examples of name types that in general are **not** capitalized. Only capitalized instances will receive a category by our system.

Object type	**Example**
Historical events, historical and geological ages	trettiårskrigen (The Thirty Year War), krystallnatten (The Crystal Night), bronsealderen (The Bronze Age), juratiden (The Jurassic Period)
Laws of the natural sciences and the judiciary, treaties	tyngdeloven (the law of gravity), barnerettskonvensjonen (The Children's Rights Convention)
Ethnic groups, languages, members of organizations, animal and plant species, illnesses	norsk (Norwegian), katolikker (Catholics), tulipan (tulip)
Days of the week, months, seasons, holidays	mandag (Monday), januar (January), påske (Easter)
Sports and games, fabrics, wines, cheeses	rugby (rugby), sjakk (chess), brie (brie)

names Norwegian exhibits two basic patterns: either all parts of the name are capitalized, or only the first name-internal word is capitalized. The latter pattern can be punctuated by name-internal names. The person name *Jan Egeland* is an example of the first pattern, the institution name *Statens lånekasse for utdanning (Norwegian State Educational Loan Fund)* illustrates the latter. This means that in contrast to English not only functional words contained in names may be uncapitalized. This second pattern constitutes a peculiarity of Norwegian.[3] By a punctuated second pattern we mean names such as *De nederlandske Antillene (The Dutch Antilles)*, where *Antillene* is the name-internal name. It is reasonable to expect that multi-part words whose parts are not all capitalized, are more difficult to delimit than multi-part names for which all parts are capitalized. Name types differ in regards to the relative frequency of the two patterns. Above all, names of public institutions and titles of for example books adhere to the "Norwegian" pattern, where only the first word is capitalized with the exemption of name-internal names. Names of public institutions can, however, have all parts capitalized: for two-part names whose first part constitutes a name, there is a tendency for the second part of a two-part name to be capitalized (Vinje, 2004). Examples of this tendency are *Oslo Sporveier* and *Norges Bank*. Middle and family names are on the other hand consistently capitalized, while private companies tend to use the pattern where all words are capitalized. We will return to the subject of lowercase-uppercase patterns for Norwegian in the next chapter, which is on attribute selection. For more examples on lowercase-uppercase in Norwegian, see Vinje (2004).

Now one thing is conventions, another thing is actual use. While for example weekdays and months are consistently uncapitalized, several name types that are listed as uncapitalized, tend to be used both capitalized and uncapitalized: examples are names of historical events such as *Vinterkrigen (The Winter War)*, *Orangerevolusjonen* (The Orange Revolution), or plants, *Vill Yams (Wild Yams)*. The rule mentioned above, that the institution should be capitalized while the person holding the office should not, as in *Fylkesmannen*, was new to this author. A further subtlety are Arabic names, a group of names often mentioned in the news at the present time. They are in many cases written sometimes capitalized, other times not. Such examples are: *al-Qaida*, *al-Jazeera* and *bin Laden*.

In spite of the above observations the vast majority of Norwegian names are capitalized. Uncapitalized names are, in the context of our work, ignored, as we rely on automatic name detection.

[3]The same pattern is found in Swedish.

We now turn to describing the corpus annotated for name category.

5.2 The Documents

Maximum entropy modeling constitutes a supervised learning technique. No name-category annotated data existed for Norwegian, so we made a 230 000 token corpus. We adopted an existing collection of texts that had not been especially assembled for the purpose of serving as test and training data for proper name classification. As we will see at the end of this chapter, some of the semantic categories we use are very scarce in our corpus.

Documents are either excerpts from contemporary fiction or from a cross-section of Norwegian newspapers and magazines. Contemporary fiction constitutes between a fourth and a fifth of the corpus. The corpus is made up by ten excerpts of five thousand words each from ten different novels published in the mid 1990s. The test data of each fold comprises one such excerpt. Texts were taken from "quality" newspapers, the tabloid press, papers aiming at both a nationwide and more local audience, women's magazines, and magazines for special-interest groups such as motorists etc. These too are from the mid-1990s. For more details on the texts constituting the corpus see Appendix A. In the previous chapter we stated that the partitions of the folds of the cross-validation were done keeping documents intact.

5.3 The Tagged Corpus

In Chapter 2 we saw that Named Entity Recognition may constitute one or two steps. We do the recognition in two separate steps: the proper names are first found, then classified. We could for example have used a separate maximum entropy model to detect the names, but instead chose to use an existing grammatical tagger.

The corpus was tagged with the Oslo-Bergen grammatical tagger, see Johannessen et al. (2000a) or Johannessen et al. (2000b) for a more detailed description of the tagger.[4] The tagger, including the module that finds the names, is rule-based in the form of Constraint Grammar. We have already touched upon a system that employs constraint grammar, in that one of the two name classifiers for Norwegian employs this method. With Constraint Grammar more than one reading may be left after disambiguation. Constraint Grammar represents the careful strategy: potentially correct readings

[4] A description and demo of the tagger is found at http://omilia.uio.no/obt/

are kept. This is the strength of the Constraint Grammar framework, but it comes at the cost of ambiguity, see Karlsson et al. (1995) for a more detailed description of Constraint Grammar.

The Oslo-Bergen tagger is known to have excellent recall, but retains some level of ambiguity. Readings assigned are rarely faulty, but there are often more than one reading. Highly common ambiguities involve gender and number, so that in many cases the lemma and grammatical category are shared by several readings. The order of the readings is in the case of our tagger not fixed but follows declining frequency: the frequency is derived from a gold standard corpus and equals the linear combination of the word frequency, the grammatical category frequency and the grammatical category-bigram frequency.

In order to illustrate what the tagger output looks like, we return to the example of the previous chapter, ie example (9):

(9) Han nevner Tyrkia som eksempel.
he mentions Turkey as example.
He gives Turkey as an example.

The equivalent tagged sentence is shown in Figure 5.1.

```
"<Han>"
        "han" pron pers 3 mask ent hum nom @subj
"<nevner>"
        "nevne" verb pres tr1 tr2 @fv
"<Tyrkia>"
        "Tyrkia" subst prop &st* @obj <sted><org>
"<som>"
        "som" prep @adv
"<eksempel>"
        "eksempel" subst noeyt appell ub fl @<p-utfyll
        "eksempel" subst noeyt appell ub ent @<p-utfyll
"<.>"
        "$. " clb <punkt><<<
```

Figure 5.1: Example sentence of the annotated corpus.

First the word form is given enclosed by "<>" as with "<Han>". The readings of the word form are indented, and each reading occupies one line.

The reading assigned to "<Han>" is *"han" pron pers 3 mask ent hum nom @subj*. Here it is only the common noun 'eksempel' which has two readings: it is ambiguous in number. A reading first gives the lemma enclosed in " ", followed by the morpho-syntactic information. Each reading will include a syntactic tag marked by @. The *subst prop*, indicating a name, in the case of actual names entails a manually assigned name category, indicated by &*: *Tyrkia* is here given the location tag. While all other information is derived from the tagger, the name category has been manually assigned. The <sted> <org> in the reading belonging to *Tyrkia* which translates as <location> <organization> means that *Tyrkia* is found on a list of countries.[5] As explained in section 6.1.6 a name's presence on a name list is encoded directly in the tagger output. Finally, tags, as said in connection with the description of the Norwegian rule-based classifier, include no semantic information.

Common ambiguities in the tagged output which involve names are the common noun/proper name ambiguity, due to the sentence-initial position of the name as in (10), the point here being that *Skagen* may also represent a common noun in determinative singular as shown in Figure 5.3.

(10) *Skagen* eller *Fredrikshavn*?
Skagen or *Fredrikshavn*?

"<Skagen>"
"skage" subst mask appell be ent @obj
"Skagen" subst prop @obj &*st*⋆ <sted>
"<eller>"
"eller" konj @kon
"<Frederikshavn>"
"Frederikshavn" subst prop @obj &*st*⋆ <sted>
"<?>"
"?" clb <spm><<<

Figure 5.2: The sentence-initial name *Skagen* is assigned both a common-name and a proper-name reading.

Another common ambiguity involving names is the genitive/non-genitive

[5]The genitive form of a name found on a list does not receive list tags, hence the genitive of *Tyrkia* would not carry such tags. We made the choice in the belief that errors would result from letting multi-part names inherit the list tag of a genitive part.

of a proper name. This happens with rare names that end with an s or rare names in genitive as in (11). The tagged equivalent is shown in Figure 5.3.

(11) *Mijailovics* forsvarer *Peter Althin* vil foreløpig ikke
Mijailovics defence attorney *Peter Althin* does currently not
uttale seg om dette.
want to comment on this.

"<Mijailovics>"
"Mijailovic" subst prop gen @obj @subj @subst> &*pe**
"Mijailovics" subst prop @obj @subj @subst> &*pe**
"<forsvarer>"
"forsvare" verb pres tr1 r14 @fv
"forsvarer" subst appell mask ub ent @obj @subj @tittel @i-obj
"<Peter Althin>"
"Peter Althin" subst prop @subj <person> &*pe**
"<vil>"
"ville" verb pres tr1 tr2 @fv
"<foreløpig>"
"foreløpig" adj nøyt ub ent pos @adv>
"<ikke>"
"ikke" adv @adv
"<uttale>"
"uttale" verb inf tr1 tr2 @iv
"<seg>"
"seg" pron akk refl ent/fl @obj
"<om>"
"om" prep @adv
"<dette>"
"dette" pron nøyt ent pers 3 @p-utfyll
"<.>"
"$." clb <<<<punkt>

Figure 5.3: The family name *Mijailovics* is ambiguous due to the final s and the rareness of the name.

It should also be noted that the tagger in the case of multi-part names only assigns a tag to the name as a whole and not to the different parts. This can be seen in *Peter Althin* of Figure 5.3.

5.3.1 Perfect Name Detection

Our corpus consists of two parts: half of the tagged corpus is correct for all tags as it served as gold standard to evaluate the tagger's performance. The second part is text tagged by the version of the Oslo-Bergen tagger of January 1st 2001. At this point the measures to detect multi-word name and names in the sentence-initial position were underway. These measures are described at length in the next section.

We chose to override the tagger by marking up as proper names the actual proper names of the corpus for the following three reasons. Firstly, many instances of multi-word and sentence-initial names were undetected in the tagged output. If we had not corrected the tagger output relative to the proper-name tag, we would have examined a classifier for mainly single-word names and sentence-internal names. Secondly, we are using a supervised learning method, which means that we need to manually assign a unique tag to each name. We found the assignment of a unique category to be problematic for cases where the word form does not represent an actual name. Many sentence-initial words that were incorrectly tagged as proper names were clearly bugs. In such cases there is no natural choice of category. In cases were the inflected word form is also a name, but is not so in the actual case, we could imagine assigning a particular category.

(12) —*Per idag* er det ikke mange HIV positive som velger å
—*at the present* are there not many HIV-positive who choose to
få barn.
have children.

—There are presently not many HIV-positive (people) who choose to have children.

While *Per* is a common first name, this is not its meaning in (12). We could imagine assigning the person category to *Per*. In fact, had the sentence and context been slightly different *Per* could have been a person name. A further argument for manual correction is that the tagger is the only name-detecting system for Norwegian. We knew that its name-detection ability was about to be improved, and it is natural to think that our system could be applied together with this improved version. It was of course at this stage unclear exactly how the improved version would perform. A downside concerning the use of our system in combination with the tagger is that our name classification system has been trained on perfect input. It is often, but not always, better to train a system on real imperfect input representing what actual input will be like.

As said before, only names that are capitalized count as names in the context of this thesis. We manually marked as proper names word strings that the tagger had not detected and removed the name tag from tokens when we considered it to be an error. We did not merely improve the tagger's precision: we not only corrected the expressions marked as name by the tagger, but read through the corpus in order to find names that the tagger had missed completely.[6] While we do not assign name categories to non-names, we allow more than one name reading of actual names to be assigned category. As shown in Figure 5.4 this eg happens in cases where there are more than one form of the name in the lexicon, such as *SV* and *Sv* which both denote the same political party. We believe that in all instances the different name readings of a name must have the same category.

"<SV s >"
"sv." fork adj gen @obj @subj
"sv." fork subst gen @det>
"SV" fork subst prop gen &*or** @det>
"Sv" fork subst prop gen &*or** @det>
"SV" subst prop gen &*or** @det>

Figure 5.4: The readings assigned to an instance of the genitive of SV. It should be noted that more than one reading has been assigned a name category. In the corpus there is no case where two readings belonging to the same token have been assigned different name categories.

We chose to mention that our corpus originally consisted of two parts because this says something about the degree of correction performed by us. As one half was perfect name-wise, errors were found only in the second half, which included bugs made by an immature method (section 5.3.2). As for the correctness of tokens for which none of the readings constitutes a name, they are perfect in the case of the one half, while as we said earlier, the tagger has an impressive recall, but high ambiguity.

There are a total of 7 532 names in the annotated corpus where the name-detection is perfect. In the previous chapter we saw how each name generates

[6]The NE-annotation was done by this author and Andra Björk Jónsdottir during the first half of 2001. Kristin Hagen at the Text Laboratory at the University of Oslo joined the discussions on markup strategies.

one line of input to the learner/classifier. In our experiments involving cross-validation we exploited the fact that only actual names have been assigned a name category, hence the input lines are generated from name readings including the name category. If a name has more than one reading which includes a name category, the top most reading is chosen. To use the presence of the name category to choose a reading is of course not possible during real application.[7]

5.3.2 The Proper-Name Detection Module

As described in the last section, it is not the tagger in the annotated data that decides which expressions constitute a name. We nevertheless describe the name detection strategies employed by the tagger, since we intend to apply our system in combination with the tagger for name type recognition of new text. The following is a description of the name-detection strategies employed by the present day tagger. The tagger initially marks as names all words that are capitalized and not sentence-initial. The lexicon includes some proper names. If a sentence-initial word is neither found in the lexicon, nor may be analyzed as a possible compound, it is marked as a proper name. Norwegian compounds do not include spaces. Entities of the form *capitalized word hyphen noun* will not be tagged as a proper name by a rule-based system, since a majority of these expressions eg *Brann-treneren* (the coach of the football team *Brann*), *Statoil-styret* (the board of *Statoil*) are not names. Compounds consisting of a proper name and a common noun as in *Rørosbanen (The Røros Railway)* are on the other hand tagged as proper names. Since not only function words that are part of a name may be uncapitalized, multi-part names in Norwegian can be expected to be harder to delimit than is the case for English. There are three ways in which the tagger subsequently decides which word or words constitutes a name: regular expressions, a document method and syntactic analysis. The first two strategies are described in Johannessen and Meurer (2002), while the syntactic analysis is described in Hagen (2003).

Regular Expressions

Regular expressions exploit the fact that names have an inner grammar (McDonald, 1996). We recall that the different parts of a multi-part name are not tagged separately in the tagger output. The example we gave was *Peter*

[7] For this reason Nøklestad (2004) during cross-validation chose not to use the presence of a name category to choose reading, but consistently preferred a name reading.

Althin. Fortunately the morphological information provided by intermediate stages of the tagger, where each word receives at least one reading, is accessible to the rules. Multi-word names such as *Universitetet i Oslo (The University of Oslo)* or *Sentralsykehuset i Akershus (The Central Hospital of Akershus)* are detected by a rule stating that a word representing a common noun in the definite form, followed by the preposition *i*, followed by a location name, constitute a name. The rule applies as *Universitetet* and *Sentralsykehuset* are common nouns in the definite form, while both *Oslo* and *Akershus* are names of locations. A further example that can be detected by a regular expression describing names consisting of noun phrases is eg *Den norske kirke (The Norwegian State Church)*.

Looking Beyond the Sentence Boundary

In Chapter 2 we saw how for example the named entity recognition system of Mikheev et al. (1999) centers on co-reference. A name may clearly occur more than once in a text, including shorter versions of a name. According to Church (2000) names are examples of semantic expressions that are characterized by what he refers to as "burstiness": while the probability of the first instance of a certain name occurring is very low, the probability of the second instance is closer to 0.5 than to the first probability. The probability of two instances of a name in a text is vastly higher than the square product of the probability of the first instance.

The idea is to use the instances which can be detected by a regular expression to find the more difficult instances including short forms of the name. The difficult case may either precede or follow the more evident case: search for the clearer case is both left and right. The search distance is kept fixed, instead of relying on the existence of text boundary marks (Johannessen and Meurer, 2002). The name *Den norske lægeforening (Norwegian Medical Association)* is easier detected in (13) than in (14), as the capitalization of *Den* marking the start of a name is ambiguous, when *Den* is also the first word of the sentence.

(13) Janne er medlem av *Den norske lægeforening*.
Janne is a member of *The Norwegian Medical Association*.

A regular expression finds the name *Den norske lægeforening* in (13). This fact is then used to identify sentence initial occurrences either preceding or following the more evident case as in

(14) *Den norske lægeforening* velger sitt styre for to år.
The Norwegian Medical Association elects its board for two years.

In the case of names that follow the pattern of *Den norske lægeforening*, the phrasal head in the definite form often serves as a short version of the longer name. The tagger therefore also finds the phrasal head of the name and recognizes this head in definite form as head *Lægeforeningen*. This method of searching beyond the sentence boundary is influenced by Mikheev et al. (1999). The procedure used by the tagger is less elaborate than that of Mikheev et al. (1999), which, in addition to heads, also recognizes strings of words as short forms, provided that the original relative order is kept intact. Hence *Kluwer Ltd* is identified as short for *Adam Kluver Ltd*, while *Ltd Kluver* is not. Mikheev et al. (1999) also recognizes acronyms. At the present time, the document method works less than perfectly.

Delimiting Names through Syntax

In general, it is difficult to establish the limits for multi-word names where only the first word is capitalized. In the following examples the Norwegian constituent order makes it difficult to establish boundaries for names whose parts are all capitalized. Syntax is used to decipher strings of names. Norwegian is a V2 language, which means that the verb is the second constituent of a main clause. If the subject is not in the canonical first position, it must follow immediately after the verb. When eg the subject is immediately followed by an indirect or a direct object, and they are both names, we get a string of names. Hence in Norwegian, names are immediate neighbors, whereas this is not the case for the English counterpart. All examples are from Hagen (2003). In examples such as (15), *Hansen* is not the family name of the woman whose first name is *Kari*.

(15) Igår ga *Kari Hansen* dokumentene.
yesterday gave *Kari Hansen* the documents.
Yesterday *Kari* gave *Hansen* the documents.

A rule which uses the fact that *Kari* has been syntactically tagged as the subject, while *Hansen* is tagged as indirect object, correctly analyzes *Kari* and *Hansen* as two separate names. In (16) the problem is to group the sequence of five first and last names, *Kari Berg Jensen Tor Hansen*, into separate names.

(16) Igår ga *Kari Berg Jensen Tor Hansen* dokumentene.
yesterday gave *Kari Berg Jensen Tor Hansen* the documents.
Yesterday *Kari Berg Jensen* gave *Tor Hansen* the documents.

The solution used by the tagger to arrive at the correct answer, that *Kari Berg Jensen* is one name while *Tor Hansen* constitutes another, is to use a rule that uses the information that *gi* (give) is a bi-transitive verb, while *Kari* and *Tor* are first names of different genders, *Kari* being the female name. Now foreign names such as *Charlie* are not gender-marked. In such cases a rule is applied which states that names in a particular context have equally many parts. Hence the four-word string *Charlie Brown Susan Smith* in (17) is correctly split into *Charlie Brown* and *Susan Smith*.

(17) Igår ga *Charlie Brown Susan Smith* dokumentene.
yesterday gave *Charlie Brown Susan Smith* the documents.
Yesterday *Charlie Brown* gave *Susan Smith* the documents.

For cases with potentially three single-word names, eg if *Charlie Brown* in (17) was referred to only by first name, there is a rule stating that the first word is a separate name, while the second and third together form a name. The motivation for this rule is that the agent, which is typically realized as the subject, is expected to be more familiar to both speaker and hearer than a benefactor in the form of an indirect object. A short version of the name is therefore used to denote the agent.

According to Hagen (2003), sentences with sequences of many name parts are rare, and the tagger often get these wrong. The name-detection module of the tagger has not been evaluated. The strategies for finding names described above are also interesting relative to the question if information useful for name detection is also relevant for determining name categories.

We believe enough has been said about name detection and therefore turn our attention to name classification: we describe our set of name categories and how they are applied.

5.4 The Semantic Name Categories

It is the capitalized names that are to be classified. Section 5.1 contains two tables, Table 5.1 and Table 5.2. It is the members of Table 5.1 and the capitalized instances of 5.2 that are to be assigned a category. The names of our corpus denote entities as diverse as railway lines (*Vestfoldbanen*), oil fields (*Oseberg*), schools, tunnels, swimming pools, military campaigns (*Enduring Freedom*), etc. Each name instance is to be labeled with exactly one name category. We consulted literature on name taxonomy by more classical

name researchers, such as Pamp (1994).[8] Two factors determined our choice of name categories: proper name categories that are conventional Named Entities and an inventory of categories useful for Internet searching.

We use six categories: PERSON, LOCATION, ORGANIZATION, EVENT, WORK and OTHER. The categories PERSON, LOCATION and ORGANIZATION, resemble the MUC-categories with the same names.[9] Our use of these categories differ from how they were used according to the MUC-markup guidelines in two ways. First, our markup places stronger emphasis on metonymy. While for example *The White House* in the MUC-scheme is always to count as an organization, we let a Norwegian equivalent *Stortinget* (The Parliament) receive the location tag, if it is the location that is highlighted. This will be explained in greater detail in the next section of this chapter. Second, while the semantics of our location and organization categories are very much the same as in the MUC, the person category of MUC is enlarged to also include names for pets and other singular animals including fictional (such as the name of the ravens of the god Odin), names of gods and names of fictional characters.

Two additional categories that were expected to be attractive for Internet search were introduced. The first category, termed WORK, covers products of the media, entertainment and the arts. Typical representatives are newspapers, magazines, TV programs, movies, paintings, music, etc. The second category, termed EVENT, covers names referring to events, both historical (*Falklandskrigen* (The Falkland War)) and cultural or sporting events such as *Tour de France* as well as weather-related events such as typhoons and hurricanes (*Katrina* (Hurricane Katrina)).

Finally, a residual category, termed OTHER, is needed so that all names can get a tag. A large group of names that are assigned the OTHER tag are product and brand names. Since such names unlike most proper names refer to a set of objects and not to a singular object, the visibility of this group is toned down by enlarging their group to also include names that do not naturally fit into any of the other five. Table 5.3 shows our name taxonomy. The six categories were not further subdivided.

[8] "We" in this context refers to Andra Björk Jónsdottir and the author. Name researcher Botolv Helleland, Institutt for navnegranskning at the University of Oslo was a contributor to the start-up phase.

[9] The MUC guidelines can be found at http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html

Table 5.3: Examples of name types that sort under the different semantic categories.

Category	**Members**
Person	individuals and families, individual animals, deities, fictional characters
Organization	companies, non-governmental organizations, governments, political parties, sports teams, public facilities such as schools, prisons, hospitals
Location	facilities, geopolitical entities (countries), geographical entities (rivers, mountains)
Work	books, musical products, media products (TV-programs, newspapers, magazines)
Event	cultural events, sports events, weather phenomena, historical events
Other	unique vehicles, brand names of wine, medicines, cars, religions

5.5 Annotation Strategy

This section focuses on metonymy and an annotation strategy we term function-over-form.[10]

Proper name classification represents word sense disambiguation. Cases of unrelated senses (homonymy) are in general unproblematic in terms of correct category: the person *Jordan* is assigned the PERSON tag, while the same named country is deemed LOCATION. In the case of homonymy, the category was assigned based on the reading of the entire document, and not just the sentence the name occurs in. The global context may indicate a different category than the more immediate context as in (18). While *Yara* is a first name in the Middle East, our document concerns the Norwegian company *Yara* selling a daughter company. Hence *Yara* is tagged ORGANIZATION and not PERSON.

(18) *Yara* (ORGANIZATION) selger datter.
Yara sells daughter.
Yara sells a daughter (company).

We illustrate our annotation strategy of function-over-form on the following two groups of names and contexts. In the first group, the administration of a country (or city or municipality) is referred to by its site as in:

- the *US* led invasion...
- *Moscow* condems...
- The *EU* president ...
- *Os kommunes (Os county's)* saving scheme ...

In these cases, the choice of category stands between LOCATION or ORGANIZATION. We chose ORGANIZATION.

The second group consisted of instances when the name of a newspaper or broadcaster can be argued to stand for the journalist as in:

- *Aftenposten* reports...
- *Aftenposten* has learnt...

[10] Jónsdottir (2003) devotes a chapter to the mark-up of the Norwegian NE corpus.

These sense pairs represent metonymy (related senses). Metonymy has a referential function: it allows us to use one entity A to stand for another entity B that is related to it (Lakoff and Johnson, 1980). Metonymy allows us to focus more specifically on certain aspects of what is being referred to. Metonymy is systematic in nature.[11] The following are some metonymic patterns taken from Lakoff and Johnson (1980):

- PRODUCER FOR PRODUCT
  He bought a *Ford*.
- CONTROLLER FOR CONTROLLED
  Bush invaded Iraq.
- THE PLACE FOR THE INSTITUTION
  Wall street is in panic.
  Paris is introducing longer skirts this season.
  Washington condemns the rampant human rights violations of Mugabe's regime.
- THE PLACE FOR THE EVENT
  Remember *Hiroshima*.

Different kinds of text exhibit different kinds of metonymy for example sports teams are frequently referred to by location names in texts on sports. Our corpus is made up by different text types.

The recognition task clearly is more difficult with the function-over-form annotation strategy than if the same name has the same category across a text. In the latter case, lists are much more powerful. At the same time, function-over-form can be more suitable than form-over-function for certain applications.

5.6 The Final Data Set

With the strategy for annotation described in the previous section a total of 7 532 names were assigned a name type. Table 5.4 gives the number of the

[11] But metonymy can also be unsystematic and hence open-ended. An example of unsystematic metonymy is the following sentence, which could be expresses by a waiter. The *ham sandwich* refers to the customer who ordered it.

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

Unsystematic metonymy is, however, far less common than systematic (Markert and Nissim, 2002, and references therein).

respective categories.

Table 5.4: The number of the different name categories in the annotated data listed in decreasing order. The most numerous category, the PERSON category, is 94 times more frequent than the least numerous category, the EVENT category.

Category	**Number of instances**
Person	3 676
Location	1 912
Organization	1 501
Other	259
Work	145
Event	39
Overall	7 532

We see that there are three very common name categories and three much less common name categories: there are in decreasing order 3 676 PERSON, 1 912 LOCATION, 1 501 ORGANIZATION, 259 OTHER, 145 WORK and finally 39 instances of EVENT. Before we move onto our next chapter which explores the usefulness of different attributes two points should be made: firstly, due to their high number a system that does well on the most common categories, namely PERSON, ORGANIZATION and LOCATION, will have a high overall performance. Secondly, due to the rareness of the categories EVENT, WORK and OTHER in the training data of each fold, attributes that are potentially useful for the classification of these three categories, will not reach their full potential. With for example only 39 instances of EVENT altogether, the number of instances present in the test data of a fold is low, hence results are more uncertain than in the case of the three most common categories.

We are now ready to present and discuss our experiments.

Chapter 6

Experiments

The topic of this chapter is the experimental investigation of different factors influencing name classification accuracy, first in terms of attributes, then with regard to model-building parameters. Attribute selection takes up the larger part of the chapter: the core question this thesis seeks to answer is which attributes are useful for an automatic classification of names in Norwegian. The smaller, final section discusses parameter optimization.

6.1 Attribute Description

We use the term attribute to refer to a characteristic of a name or a neighbor. We reserve the term feature for the indicator function that takes attribute value and (name) category as arguments. The maximum entropy model is defined in terms of features. This chapter first describes different attributes and discusses how they are implemented. The corresponding results are then reported and discussed. We explore the effect of attributes commonly used in machine learning based NER where the set of NEs is rather small and NEs are not subdivided. We also implement an attribute that captures the feature particular to Norwegian multi-part names, namely that it is only the initial part of the name that must be capitalized. All attributes are local information. The attributes are derived either from unprocessed text or from the tagged text.

We now describe the different attributes.

6.1.1 The Lexical Window

NER systems standardly use lexical windows anchored at the NE. We examine 2 x 2 different representations of the NE and its neighbors. In the case

of the first two alternatives, the NE and the neighbors are represented in the same way: either all are represented as inflected forms or all are represented as lemmas. Neighbors of the NE in question may also clearly be names. In the third and fourth alternatives, names and non-names are treated differently: in the third alternative, names are represented as inflected forms and non-names as lemmas, whereas in the fourth and final alternative, it is the other way round.

In the case of all four representations, we examine the effect of providing the system with symmetric windows of growing size, insofar as same-sentence neighbors exist. The smallest window consists only of the name, while the maximal window tried comprises the name and five neighbors on each side.

We have already encountered a window of inflected forms as we, in Chapter 4, used the encoding of a lexical window to illustrate what the input to the learner looks like:

(19) Han nevner Tyrkia som eksempel.
he mentions Turkey as example.
He gives Turkey as an example.

The differently sized windows of inflected forms centered around the name *Tyrkia* equal (20)-(23):

(20) w0=Tyrkia LOCATION

(21) w-1=nevner w0=Tyrkia w1=som LOCATION

(22) w-2=han w-1=nevner w0=Tyrkia w1=som w2=eksempel LOCATION

(23) w-2=han w-1=nevner w0=Tyrkia w1=som w2=eksempel w3=. LOCATION

In the case of (19), (23) represents the maximum window (windows of sizes 4 and 5 equal this window). Up until size two, windows are symmetric as sentence boundaries are not crossed. We further notice that the attribute corresponding to the sentence initial non-name *Han* is not capitalized and that attribute values include sentence boundary markers (w3=.).

The second representation of the lexical window amounts to replacing inflected forms with lemmas, which allows rare events to merge into less

rare events. The use of lemmas over inflected forms is a standard idea, eg the word-sense-disambiguation system for Dutch described in Gaustad (2004) uses neighbor lemmas and not word forms. The morpho-syntactic information, such as tense and number, which is missing from the lemma, ought to be of little use to the classification of the neighboring name.

In section 5.3, we reported that the tagger frequently assigns more than one reading to a token, hence there is the question as to which reading to choose the lemma from. In the case of names, we know the correct reading because of the way the annotated corpus was tagged. Hence, *Skagen* is both an NE and a proper name that neighbors *Fredrikshavn*.

(24) *Skagen* eller *Fredrikshavn*?
Skagen or *Fredrikshavn*?

"<Skagen>"
"skage" subst mask appell be ent @obj
"Skagen" subst prop @obj &*st** <sted>
"<eller>"
"eller" konj @kon
"<Frederikshavn>"
"Frederikshavn" subst prop @obj &*st** <sted>
"<?>"
"?" clb <spm><<<

Figure 6.1: Although the topmost reading of *Skagen* had the highest frequency of the two readings, we nevertheless choose the second reading, that of a name. The reason being that in the cross-validation corpus name category was only assigned to actual names.

(25) l-2=Skagen l-1=eller l0=Fredrikshavn l1=? LOCATION

If no name reading is among the readings, we choose the lemma of the topmost or single reading, since readings, as explained in the previous chapter, are ranked with the topmost reading being the most frequent in a gold standard corpus. Figure 6.2 is the corresponding tagged sentence of (26).

“<Utenfor>”
	“utenfor” prep @adv
“<Norges>”
	“Norge” subst prop gen &st* @det>
“<grenser>”
	“grense” subst mask appell ub fl @<p-utfyll
	“grense” subst fem appell ub fl @<p-utfyll
“<hadde>”
	“ha” verb pret pa1 a6 d5 r16 tr6 d6/til pa3 tr12 <aux1/perf-part> pa6 @fv
“<han>”
	“han” pron pers 3 mask ent hum nom @subj
“<stadig>”
	“stadig” adj pos noeyt ub ent @adv
“<en>”
	“en” det kvant mask ent @det>
“<trofast>”
	“trofast” adj pos m/f ub ent @adj>
“<leserskare>”
	“leserskare” subst mask appell ub ent samset @obj
“<.>”
	“$. ” clb <punkt><<<

Figure 6.2: The topmost reading is consistently chosen whenever no name reading is among the readings. Here the readings assigned to *grenser (borders)* have identical lemmas, but two possible grammatical genders. It is not uncommon for the number of different lemmas to be smaller than the number of readings.

(26) Utenfor Norges grenser hadde han stadig en trofast leserskare.
outside Norway's borders retained he a loyal readership.
Outside Norway's borders he retained a loyal readership.

In Figure 6.2 the lemma of the topmost reading assigned to *grenser (borders)* equals *grense (border)*, but in this case the lemmas of the two readings are identical. (Norwegian has three grammatical genders: feminine, masculine and neutral. Many nouns can be used both in the feminine or masculine form, while the choice does not affect its meaning. While the indefinite plural

of grense (border) is the same for both grammatical genders, *a border* translates as *en grense* in the masculine case, as *ei grense* when it is used as a feminine noun.) As said in the previous chapter, it is not uncommon for lemma and grammatical category to be the same for several of the readings assigned to a token. (27)-(32) show the differently sized lemma windows.

(27) l0=Norge LOCATION

(28) l-1=utenfor l0=Norge l1=grense LOCATION

(29) l-1=utenfor l0=Norge l1=grense l2=ha LOCATION

(30) l-1=utenfor l0=Norge l1=grense l2=ha l3=han LOCATION

(31) l-1=utenfor l0=Norge l1=grense l2=ha l3=han l4=en LOCATION

(32) l-1=utenfor l0=Norge l1=grense l2=ha l3=han l4=en l5=trofast LOCATION

A less common situation arises in Figure 6.3 where the adoption of the topmost reading of *forsvarer* results in the encoding of an incorrect attribute: the correct lemma of *forsvarer* is found in the bottom reading. Information from more than one reading can be preserved. We could, in the above case, have chosen to encode both the lemmas given for the word form *forsvarer*.

Names might differ from non-names in how they are best represented as an inflected form or as a lemma. In the third alternative the NE was therefore represented as inflected form. The neighbors were represented with their inflected form in the case of a name, and with the lemma of the first or unique reading when non-names. (33)-(38) show the different windows for the third lexical representation. To ease reading, only the attribute values are given.

(33) *Norges* LOCATION

(34) utenfor *Norges* grense LOCATION

“<Mijailovics>”
	“Mijailovic” subst prop gen @obj @subj @subst> &*pe*⋆
	“Mijailovics” subst prop @obj @subj @subst> &*pe*⋆
“<forsvarer>”
	“forsvare” verb pres tr1 r14 @fv
	“forsvarer”subst appell mask ub ent @obj @subj @tittel @i-obj
“<Peter Althin>”
	“Peter Althin” subst prop @subj <person> &*pe*⋆
“<vil>”
	“ville” verb pres tr1 tr2 @fv
“<foreløpig>”
	“foreløpig” adj nøyt ub ent pos @adv>
“<ikke>”
	“ikke” adv @adv
“<uttale>”
	“uttale” verb inf tr1 tr2 @iv
“<seg>”
	“seg” pron akk refl ent/fl @obj
“<om>”
	“om” prep @adv
“<dette>”
	“dette” pron nøyt ent pers 3 @p-utfyll
“<.>”
	“$.” clb <<<<punkt>

Figure 6.3: By chosing the lemma of the topmost reading of *forsvarer*, we actually encode incorrect infomation.

(35) utenfor *Norges* grense ha LOCATION

(36) utenfor *Norges* grense ha han LOCATION

(37) utenfor *Norges* grense ha han en LOCATION

(38) utenfor *Norges* grense ha han en trofast LOCATION

The fourth logical combination is to represent names as lemmas, whereas non-name neighbors are consistently represented as inflected forms:

(39) *Norge* LOCATION

(40) utenfor *Norge* grenser LOCATION

(41) utenfor *Norge* grenser hadde LOCATION

(42) utenfor *Norge* grenser hadde han LOCATION

(43) utenfor *Norge* grense hadde han en LOCATION

(44) utenfor *Norge* grense hadde han en trofast LOCATION

For the window sizes that resulted in the best results, we, for all four name and neighbor representations, tried removing either the left or right neighbors, thus making asymmetric windows. Results of only providing a lexical window are reported in section 6.2.2.

6.1.2 The Suffixes of the Name and its Neighbors

We again studied different sized windows anchored at the name. Suffix information of the word to be classified is standardly used in NER systems, where it has also been used for name detection.

Many common Norwegian last names end with *-sen*, as in *Hansen*. The second big trend involving Norwegian family names is that the name of the farm that the family once occupied is used as family name, as in the case of this author whose family name *Haaland* also denotes a farmstead. Because of this there are numerous suffixes such as *-land* that are shared by both family names and location names. These location names are not only limited to smaller places, but may also denote cities, provinces and countries: the country name *England (England)* and the name of a province *Oppland* have the *-land* suffix in common with the family name.

Clearly a multi-member name will often comprise a part that states what kind of entity the name refers to, ie if the name refers to a prison, sports club, school, church, museum, etc. Norwegian, unlike English, forms compounds without spaces as for example the numerous compounds of *hus* (house, hall) illustrate: *sykehus (hospital)* literally *sickhouse*, *konserthus (concert hall)*, *bygdehus (village hall)*, *kulturhus (culture hall)*, *parkeringshus (multi-storey car park)*, *rådhus (city hall)*. This is seen in names such as for example *Ullevål universitetssykehus (Ullevål University Hospital)*, *Bærum kulturhus (Bærum Culture Hall)*, *Vika parkeringshus (Vika Parking House)* and *Oslo rådhus (Oslo City Hall)*. Likewise there might be compounds in the immediate context of the name that indicate its category, such as compounds with, for example, *leder (leader)* and *by (town)* as base words.

As with the lexical windows, the suffixes of the name and the neighbors can be derived from either the corresponding lemmas, inflected forms or a mix of the two: we chose to focus on the inflected forms. Example (45) below is an example of a symmetric suffix window of size two, where the length of the suffix equals three:

(45) suf-2=han suf-1=ner suf0=kia suf1=som suf2=pel LOCATION

The corresponding example sentence is (19), while the same-sized window of inflected forms is found in (22). In the case of (45), none of the attribute values represent "classical" suffixes such as *-sen* which is clearly a good indication of a family name.

Suffixes are implemented by us as case-preserving which means that the location names *Fusa* and *USA* do not have a common suffix of length three. The suffix attribute is completely contained in the corresponding lemma or inflected form, depending on how the suffix is implemented: the suffix *usa* is fully contained in the the name *Fusa*, but the presence of such overlapping attributes does not pose a problem for the maximum entropy model. As with

the lexical windows, the cells of the suffix window anchored at the name can have neighbors that equal a clause boundary token.

The accuracy obtained from providing only suffix is reported in section 6.2.3.

6.1.3 The Grammatical Category of the Preceding Word

We record the grammatical category (POS) for different sized symmetric windows anchored at the name. For example, in the nominal phrase *Mijailovics forsvarer Peter Althin*, *forsvarer* which immediately precedes the proper name *Peter Althin* has two readings as seen in Figure 6.4.

"<forsvarer>"
    "forsvare" verb pres tr1 r14 @fv
    "forsvarer" subst appell mask ub ent @obj @subj @tittel @i-obj

Figure 6.4: This figure shows the two readings of 'forsvarer'.

We chose to implement the grammatical category of the word preceding the name in two versions: in the first version the POS of the word preceding the name is set equal to the topmost reading assigned to this word. Illustrated in the case where *forsvarer* is the word preceding the person name *Peter Althin*, in this version, the attribute of which encodes the grammatical category equals:

(46) pos-1=verb PERSON

In the second version the grammatical categories of the two or three topmost readings are encoded. The attribute of the same example now equals:

(47) pospre=verb-subst PERSON

The grammatical categories of the name-internal words can be expected to be more useful for an automatic classification than the grammatical category of neighbors. Similar to the regular expressions used by the tagger to group words into multi-word names (see section 5.3.2), the same information

could serve to identify the name category. Patterns that apply to names like *Den norske opera* or *Røde kors* which represent determiner-adjective-noun or adjective-noun respectively are typical of organization names. They at least do not represent typical person names. Furthermore, one would expect verbs to be found only in the titles of names of novels and films which sort under the WORK category. The problem though for this feature to work, is that the tagger has been constructed as to aggressively find the verb as the central element of its morphological analysis, so that a verb will rarely be interpreted as a part of a proper name, but rather as the verb of the clause. We did not, however, implement this feature due to a time-constraint combined with the fact that information on the grammatical categories of the name-internal parts are missing in the tagger output.

Grammatical category is a more abstract entity and obviously less sparse than for example suffix, and we are interested in finding out how more abstract attributes fare compared to the more specific attributes. The results from providing only this attribute are topic of section 6.2.4.

6.1.4 Uppercase-Lowercase Attributes of the Name

The MUC-7 showed that capitalization patterns were useful for recognizing names of persons, locations and organizations. Capitalization patterns have been in standard use for NER since. We start by defining an attribute which captures acronyms. We then define a Norwegian-specific attribute concerning initial capitalization of the name-internal words. Rules governing the use of capital letters in Norwegian names were presented in the previous chapter. We explained that only the first part of a Norwegian multi-part name is necessarily capitalized, but did not say anything about acronyms.

Acronyms

The majority of Norwegian acronyms are written all uppercase such as *EU*, the acronym of *The European Union* while exceptions are for example: *DnB (Den norske bank) (The Norwegian Bank).* According to Vinje (2004) acronyms that are pronounced not as a sequence of the name of the letters, but as ordinary words (in that it is the sound associated with the letter that is pronounced) may have the spelling of ordinary names: they may equally well occur with only initial capitalization as in for example *Nato/NATO, Opec/OPEC, Norad/NORAD or Obos/OBOS.* Additionally, if an acronym is no longer experienced as an acronym, it might be written all lowercase:

hence the form *aids* is used alongside both *AIDS* and *Aids* and *hiv* as an alternative to *HIV* and *Hiv*.

Acronyms obviously often replace the longer version of names. This practice seems above all to apply to organization names: *WHO* is used instead of *World Health Organization*, *EU* instead of *European Union* etc. But acronyms may also apply to people, such as *JFK* for *John F Kennedy*, to illnesses such as *AIDS* and *CJD (Creutzfeldt-Jakob Disease)* to locations, *PNG* for *Papua New Guinea*, *LA* for *Los Angeles*, for newspapers, *VG* for *Verdens Gang*, etc.

Our attribute is only to capture the all-uppercase acronym. Furthermore, we want the acronym attribute to capture true acronyms, ie we would like to exclude names that happen to be written in uppercase letters. The following sentence from the annotated corpus illustrates this point.

(48) Av JØRN J. FREMSTAD OSLO: Skihopperen Øyvind Berg kan
by Jørn J. Fremstad Oslo: the ski jumper Øyvind Berg can
tenke seg å ofre helsa for idretten han elsker.
imagine to sacrifice his health for the sport he loves.

By JØRN J. FREMSTAD OSLO: the ski jumper Øyvind Berg can imagine to sacrifice his health for the sport he loves.

Here, both the name of the journalist, *Jørn J. Fremstad*, and the place he reports from, *Oslo*, are given in uppercase. We would ideally like neither *Oslo* nor *Jørn J. Fremstad* to count as acronyms. It is natural to define acronyms as single-word names, so the journalist's name can be excluded, but this still leaves us with the problem of OSLO. Would it help to define acronym not as an attribute applying to the inflected form, but as applying to the name's lemma?

We now need to bring in how uppercase-lowercase in the inflected word form relates to that of the lemma assigned by the tagger. If the equivalent word form is found in the lexicon, the tagger does not treat uppercase-lowercase as a fixed feature of a word form. This makes the tagger flexible to different formatting practices.

Since *Oslo* is found in the lexicon this is the lemma of the unique reading assigned to OSLO, see Figure 6.5. The name *JØRN J. FREMSTAD* is uppercase also as a lemma, since it is not found in the lexicon, but as we have seen this example could be excluded as it is not a single-word name.

In the case of OSLO, it would be more useful to use the lemma and not the inflected form. In some instances using the lemma or the inflected form would give the same result: the organization name *NHO* is identical

```
"<Av>"
    "av" prep @adv
"<JØRN J. FREMSTAD>"
    "JØRN J. FREMSTAD" subst prop @<p-utfyll &pe* <person> < *stad>
"<OSLO>"
    "Oslo" subst prop @obj &st*
"<:>"
```

Figure 6.5: Illustration of lowercase-uppercase in inflected form and lemma. The person name *JØRN J. FREMSTAD* is unchanged as a lemma, while the lemma of *OSLO* is *Oslo*. The change of the latter is due to *Oslo* being in the lexicon employed by the tagger.

as inflected form and lemma, since only the all-uppercase version *NHO* is found in the lexicon. But are there cases in which we may lose acronyms by using the lemma and not the inflected form? A case where we might lose out on an acronym is *SV*, which denotes a political party. There exist different uppercase-lowercase versions of this name in the lexicon: *SV* and *Sv*, see Figure 6.6.

```
"<SV s >"
    "sv." fork adj gen @obj @subj
    "sv." fork subst gen @det>
    "SV" fork subst prop gen &or* @det>
    "Sv" fork subst prop gen &or* @det>
    "SV" subst prop gen &or* @det>
```

Figure 6.6: Both *SV* and *Sv* are found in the lexicon.

The conclusion of the above reflections is that the acronym attribute only applies to single-word names. It is a priori not clear if implementing an acronym attribute as to apply to the inflected form or to the lemma gives the best results. The choice might not matter. We, therefore, implemented two versions of this attribute: one applies to the inflected form of the name, the second to the lemma. When acronym attribute applies to the inflected form of the name, we accept lowercase genitive s, so that *EUs* counts as an

acronym. Results are found in section 6.2.5.

Distribution of Initial Capitalization of the Name-internal Words

Norwegian demands only that the first word of a multi-word name must have initial capitalization, hence there are several patterns concerning the initial capitalization of the name-internal words. Name types differ in relation to the distribution of the different name patterns: the names of public institutions are characterized by capitalization of only the first word of a multi-word name, as in: *Den norske opera (The Norwegian Opera Company)* and *Tåsen skole (Tåsen School)*. If such a name contains a proper name that is not the first word of the name, more than the first word will be capitalized, as in *Universitetet i Oslo*. Names of Norwegian books exhibit the same pattern *Kvinnen som kledde seg naken for sin elskede* (*The Woman who Undressed for her Beloved*). Here too, if there is a name within the name, there will be more than one capitalized word, as in *Alberte og Jacob(Alberte and Jacob)*. Person names have with only few exceptions all parts that are capitalized. The rare exceptions are foreign names like *ter Doest, von Koss.* Our implementation categorized multi word names as being one of three types: are all, some or none of the non-first words capitalized.[1] Two word names naturally only come in two types, the second word is either capitalized or not:

(49) *Peter Althin*; cappattern=all PERSON
Alberte og Jacob; cappattern=some WORK
Den norske opera; cappattern=none ORGANIZATION

The results for this attribute alone are reported in section 6.2.6.

6.1.5 Digits

MUC-7 participants used patterns for digits to recognize types of numeric expressions, for example an expression of the form of four digits might represent a year. Our system is not to recognize numeric expressions such as currency. Still, there is a possibility that different name types contain numbers with different frequencies. The digit attribute simply records the presence of a digit.

[1] Nøklestad (2004) uses a twofold partition, we a threefold partition of patterns, as this gave best results in our system.

6.1.6 Lists of Names

We did not experiment with different sizes or different criteria for including names in a list. Each list remained the same during our experiments. Our lists stem from four sources: firstly, names of different categories were already in the lexicon employed by the tagger prior to our addition of external lists. Secondly, our person-name lists were made using the lists of male and female first names as well as family names derived from The National Statistics Agency (Statistisk sentralbyrå). The National Statistics Agency lists all first and family names that in 2003 have at least 200 bearers of the name. Location-names were adopted from the gazetteer of The Norwegian Language Council (Norsk språkråd). Its list of location names aims to serve as guidance for users of both the Norwegian written standards as to how international geographic names are best written for both bokmål and nynorsk. We ignored the nynorsk version.[2] Names of countries or cities found in this list were, since they have administrations, also included in the list of organizations. Finally, the names of for example important Norwegian newspapers were collected by us.[3] Their names were not only included in the list of WORK names, but also in the organization name list.

Listed names total some 13 200 names. Table 6.1 gives the number of names of each type. The number of names on each list vary greatly. The list of LOCATION names represents the longest list: half of all the names listed are LOCATION names. The list of EVENT names represents the other extreme, with only 16 such names on the list. The relative number of names listed resembles the relative number of the names in the cross validation corpus (Table 5.4). In the annotated corpus too the three categories WORK, EVENT and OTHER were much less numerous than the other three categories. Because of the mark-up strategy, some names are on more than one list, for example newspapers can be both WORK and ORGANIZATION. Name list information is encoded in the tagger output.[4] If we return to the tagged examples of the chapter on the cross-validation data, this means that *Mijailovic* was found on neither list as it lacks an element enclosed by <>. In accordance with the mark-up strategy used for name category annotation, country names receive both organization and location tags from the list.

[2]Since our system is for bokmål, we removed the nynorsk equivalent from the gazetteer. For some locations in addition to the standard bokmål form, alternative bokmål spellings, in many cases older forms of this name as well as the name in the native language of the location are given.

[3]By Kristin Hagen and Andra B. Jónsdottir, both at the University of Oslo.

[4]Paul Meurer, University of Bergen, included the lists in the system.

Table 6.1: The number of the different name types on each list. The lists of PERSON names and LOCATION names respectively are the most exhaustive. The lists for WORK, EVENT and OTHER names are much smaller, with the EVENT list being very short.

Category	**Number of Instances Listed**
Person	5 486
Location	6 690
Organization	734
Work	149
Other	138
Event	16
Total	13 213

Country names receive two list tags, whereas small places were given only the location tag, arguing that smaller places are less often referred to with an administration. This means that a name that was listed as country receives the double tagging, as in the case of *Tyrkia*, whose tags are <sted><org>, the equivalent of location and organization. *Skagen* and *Fredrikshavn* are identified as location names, but not as country names, hence they receive a single location tag, <sted>. Media corporations/newspapers receive both work and organization tags.

Multi-word names inherit the list tags of their parts, so that *Peter Althin* will get the person list tag via the common first name *Peter*, while the family name, *Althin*, is not common enough to be on the list. This strategy sometimes fails: the Finnmark division of the *Tine* company *Tine Finnmark* is assigned the name list tags of person and location since *Tine* is a common female first name, and *Finnmark* denotes a Norwegian province. It likewise makes little sense that the book *"Jødenes historie i Norge gjennom 300 år" (Three Centuries of Jewish Life in Norway)* inherits the name list tags of location and organization from *Norge* (Norway). Names in the genitive form do not receive any name list tag, as it was thought that multi-part names involving a genitive frequently will belong to a different category than the name in genitive: the book containing all laws passed in Norway, *Norges Lover*, is of a different category than *Norge (Norway)*. The effect of name lists by themselves is reported in section 6.2.7.

6.1.7 Candidate Attributes

Syntactic information and co-reference attributes represent possible features which we did not implement. The tagger employs syntactic information to delimit names (section 5.3.2). There existed no full parser for Norwegian at the time when most of thesis work took place.[5] Velldal (2003) has implemented a shallow parser that finds syntactic relations. Nøklestad (2004) used this parser with slight modifications to find three syntactic relations: the proper name as a complement of a preposition, or as a subject or an object of a verb.

Mikheev et al. (1998) and Borthwick (1999), who apply maximum entropy models to NER, reported that the use of co-reference resolution improved performance (section 2.3). Unlike our system, the two systems both find and classify the proper names. The module of the tagger that detects proper names used a document method (section 5.3.2). In section 5.5, we explained how we let function-over-form determine the category of a name. As a result, annotation was in accordance with Krovetz (1998) who states that related senses such as metonymy typically result in more-than-one-sense (tag)-per discourse (text). The corpus, however, was divided into test and training data by assigning whole documents to either of the two. Training and testing is therefore never on different parts of the same document, hence co-reference is only cross-document.

With this description of candidate features, the description of the different attributes has come to an end.

6.2 Results for Single Attributes

Results are presented in three steps. First, results for the separate attributes are reported. We then move on to combining the most important attribute with an additional attribute. Finally, attributes are then added one by one, starting with the most useful. When we talk about a single attribute, we mean one attribute group.

[5] A project for building a parser for Norwegian bokmål started at the Text Laboratory of the University of Oslo end of year 2003. It will use the Oslo-Bergen tagger and a component from Denmark to build the possible trees. Maximum entropy modeling will be used to rank the different parse trees for each sentence. There are also LFG and HPSG parsers that could have been used had the project started now (2007), developed at the universities of Bergen and Trondheim, Norway.

6.2.1 A Baseline Classifier

Before examining the results corresponding to the different attributes, we record results of assigning the PERSON tag to every NE. PERSON names are the most numerous in the annotated corpus: 3 676 of a total of 7 532 are PERSON names. The model that results from no other information than the distribution of the different categories in the annotated corpus assigns the PERSON category to each name.

Table 6.2: The baseline model: each name is assigned the PERSON tag.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Person	100.00	(0.00)	48.10	(5.35)	65.19	(4.40)
Organization	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Location	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall			48.10	(5.35)		

The three accuracy measures recall, precision and F-measure are defined in section 4.3. All accuracy results are given as the empirical mean of the ten runs followed by the standard deviation. Table 6.2 shows the results of assigning the PERSON category to every NE. Starting with the leftmost column, the mean recall of PERSON is 100 percentage points, while it is 0.00 for all other categories. Furthermore, the standard deviation of recall is consistently 0.00 as all runs are identical.

In the case of precision, the mean value for PERSON is 48.10, while standard deviation is 5.35 (the middle cell of the top row). Table 6.3 gives the relevant numbers for computing the mean and standard deviation for precision and F-measure of the PERSON category. We use this classifier to illustrate how we define these accuracy measures in the case of division by zero: for all other categories, the category is never assigned, hence precision for every run is by the definition in section 4.3 undefined as we are attempting division by zero. We let precision equal zero, but could equally well have set it to 100 as there are also no false positives. The standard deviation is evidently zero as all runs are identical.

F is the harmonic mean, which is zero as soon as one of the components is zero.[6] Hence, the F-measure of categories whose recall is zero is also zero (rightmost column minus the top cell). The initial definition means that it is undefined if both P and R equal zero. It can also be regarded as undefined, once either P or R is undefined. The overall mean equals 48.10 (5.35).

Table 6.3: The relevant numbers for calculating precision and F-measure for PERSON. The left column shows the number of PERSON names in each fold, while the second column contains the total number of names in the same fold. The two rightmost columns show precision and F-measure for the different folds.

Fold	**Person**	**Total**	**Recall**	**Precision**	**F-measure**
1	383	695	100.00	55.11	71.06
2	287	668	100.00	42.96	60.10
3	406	916	100.00	44.32	61.42
4	362	802	100.00	45.14	62.20
5	592	1052	100.00	56.27	72.02
6	209	497	100.00	42.05	63.39
7	520	950	100.00	54.74	70.75
8	296	654	100.00	45.26	62.32
9	286	600	100.00	47.67	64.56
10	329	698	100.00	47.13	64.07

Concerning the entries equal to zero in the tables of this chapter recording the results of the different categories: the way the accuracy measures have been defined, we do not separate between zero as nominator and as denominator. All categories are represented in the test data of each fold, hence recall does not involve division by zero, and is therefore always defined.

[6]Remember $F = \frac{2PR}{P+R}$

6.2.2 Results for the Lexical Windows

We now leave the classifier empty of attributes behind. Section 6.1.1 described four different lexical representations of the name and its same-sentence neighbors. Tables 6.4 and 6.5 show results for different symmetric window sizes and alternative representations of the name and its neighbors. Columns correspond to different window sizes, rows to the different representations. The first row gives results for when the name and neighbors are represented as inflected forms. The second row shows results for when lemmas replace inflected forms, while the third shows results for when names and non-names are represented differently: a lemma represents non-names, the inflected form the name (mixed 1). The bottom row represents the alternative where names, whether the NE or a neighbor, are represented by a lemma, non-name neighbors as inflected forms (mixed 2).

Table 6.4: The results for the four different representations of the name and neighbors are shown over two tables. This first table shows results for the name alone or with its immediate left and right neighbors. Regardless of representation, it is dramatically better to use the name alone (leftmost column) than with its closest left and right neighbors.

	Name only		**1**	
Representation	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Inflected forms	34.14	(7.76)	22.36	(3.71)
Lemmas	33.18	(8.02)	25.96	(3.41)
Mixed 1	34.14	(7.76)	23.05	(3.27)
Mixed 2	33.18	(8.02)	23.63	(3.43)

Table 6.4 demonstrates how regardless of representation it is considerably better to use the name alone, than to use the name and its closest left and right neighbor: performance drops from the mid-thirties to mid-twenties. All results of this table are considerably worse than simply assigning the PERSON tag to every NE. In the case of the standard deviation s, the standard deviation of the name-only (leftmost column of table 6.4) is for all representations much bigger than for the windows with any number of neighbors: for name-only it is 8 or close to 8, whereas standard deviation elsewhere varies between 1.8 and 3.7.

Table 6.5 shows results for the four representations in the same order as

Table 6.5: Continued from above: results for the four different representations of the name and neighbors. (The internal order of the representations is unchanged.) In addition to the name 2, 3, 4 or 5 left and right neighbors are provided. The accuracies shown here are the double of using only the name, and three times as high as using the name with one left and right neighbor (previous table). Adding more neighbors to a symmetric window of size two yields only smaller changes in accuracy.

2		**3**		**4**		**5**	
$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
66.18	(3.24)	67.77	(2.69)	67.62	(2.10)	66.73	(1.90)
69.89	**(2.09)**	**69.65**	**(2.39)**	**69.56**	**(2.10)**	68.82	(2.18)
67.41	(2.58)	68.11	(2.25)	67.78	(1.93)	67.32	(1.83)
68.99	(2.63)	**69.64**	**(2.88)**	68.77	(2.33)	68.08	(2.36)

before. The number of neighbors to each side of the name varies from two through five. Each row has in common that the results for two, three or four neighbors are very similar, while the result for five is slightly lower. The results are in all cases roughly double of those achieved by the name alone.

The dramatic drop in performance from the name alone to the name plus the left and right neighbor and again the very steep rise from one to two neighbors is striking. So does this mean that the most immediate left and right neighbors are harmful, and that ignoring them would give better results for the larger windows? In the case of inflected forms, mean overall performance of the name and the second closest neighbors is, however, only 19.07. This is even lower than the result of 22.36 for the name and the closest neighbors, which is in agreement with the intuition that the better clues for category are found closest to the name. If we examine results for each fold, we see that in each fold the number of times the different categories are assigned is roughly the same: the EVENT, OTHER and WORK categories are assigned about as often as, for example, the PERSON category in spite of the latter being vastly more frequent in the training data. This corresponds to the definition of the maximum entropy model: no information (here, in the form of poor information) yields a close to uniform probability distribution. The shape of the performance curve from 0 to 1 to 2 neighbors exhibits an intrinsic property of the maximum entropy model: weak features can in conjunction yield a very high probability of the correct category. The latter

observation is also made on page 37 of Borthwick (1999).

The results for lemmas are consistently somewhat higher than for the other three representations, with one exception (in the case of the lone name it is better to use the inflected form as in the top and bottom row, than the lemma). A cluster of top results are shown in bold: these include the results of the *lemma* windows of size 2, 3 and 4 in addition to the second of the mixed representations with a window size of 3. The highest score 69.89 (2.09) corresponds to the lemma window of size 2, while a lemma window of size 4 has the lowest score of these four. But a McNemar's test let us conclude that the highest scoring classifier which employs a symmetric lemma window of size two is no better than the lowest-scoring of the four results: $\chi^2 = 0.39, p = 0.53217$. Hence we conclude that the classifiers corresponding to the four top-results are actually equally good.

The lemma windows of size 2 represents the smallest window which yielded optimal accuracy. We were interested in whether there was any superfluous information, that is what the effect of removing neighbors would be. We also know that the result for the name and one left and right neighbor to be disastrous. Table 6.6 shows the results of removing left or right neighbors from a symmetric lemma window of size two. Removing either the second neighbor to the left or right clearly does not represent as good an option as the symmetric window of size two. The removal of the (second) neighbor on the left corresponds to an F-measure of 54.74 (4.08). The removal of the second neighbor to the right, with 61.09 (3.01), is less harmful.

Table 6.6: Symmetric and asymmetric windows of lemmas: a symmetric window of the name and its two neighbors is the largest window. We here use bold types to distinguish between symmetric (shown in bold) and asymmetric alternatives. While a symmetric window of size two represents an optimal lexical window, removing a second neighbor clearly does not.

	0 right		**1 right**		**2 right**	
	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
0 left	**33.18**	**(8.02)**	21.37	(4.67)	16.03	(3.69)
1 left	25.26	(5.08)	**25.96**	**(3.41)**	54.74	(4.08)
2 left	20.95	(3.45)	61.09	(3.01)	**69.89**	**(2.09)**

Table 6.7: The name and two neighbors both left and right, all represented as lemmas.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	83.08	(3.39)	76.63	(4.48)	79.60	(2.28)
Organization	53.11	(6.20)	65.61	(8.58)	58.32	(5.49)
Location	71.69	(4.67)	67.31	(7.79)	69.11	(4.92)
Event	4.50	(9.56)	5.00	(10.54)	4.72	(9.98)
Work	15.08	(9.99)	21.27	(13.66)	16.55	(9.86)
Other	11.78	(10.08)	16.75	(13.30)	12.55	(9.33)
Overall			69.89	(2.09)		

Table 6.7 shows results for a lemma window of five cells, the highest scoring alternative. It should be noted that precision, recall and F-measure are necessarily identical in the overall results.[7] The highest means are achieved for the PERSON names (top row), while the worst results are for the EVENT category. The difference for the two categories is dramatic: the F-measure of PERSON is 79.60, for EVENT only 4.72. The results of the three top rows with the results of PERSON, ORGANIZATION and LOCATION are dramatically better than for the three remaining categories. Moreover, the standard deviation s of the latter three categories is much higher than for the first mentioned categories. This partly reflects the fact that the annotated data comprise more instances of PERSON, ORGANIZATION and LOCATION names, than of the remaining three categories. The results of the top three are spread out: in the case of Table 6.7, the F-measure drops by around 10 from PERSON to LOCATION and another 10 from LOCATION to ORGANIZATION.

[7]We saw in Chapter 4 that F is defined as $F = \frac{2PR}{R+P}$. In the case for which $R = P$, $F = \frac{2PP}{P+P} = \frac{2P^2}{2P} = P$. We get the symmetric result if instead of substituting for R we substitute for P.

6.2.3 Suffix Windows as the Only Attribute

As windows of suffixes clearly overlap with lexical windows, we examine results for windows of suffixes relative to lexical windows. This attribute is described in section 6.1.2. The final row termed *complete* shows results for the window of inflected forms. The previous section showed that lemma windows were more efficient than windows of inflected forms, but we are at this point interested in the comparison. Tables 6.8 and 6.9 show mean F-measures for suffix lengths of three, five and unrestricted, respectively. As in the previous section, symmetric window sizes vary from zero through four neighbors. Interestingly, in the case of suffixes, there is not the drop in performance from zero to one neighbor as lexical windows. In Table 6.8, a suffix of length five (the middle row) yields a higher empirical mean than the same sized *complete* window. In the case of a window size of three neighbors, five last letters comes very close to the unconstrained case.

Table 6.8: Suffix of length three or five of the inflected forms for window sizes of zero, one or two neighbors on the left and right.

	Name Only		**±1**		**±2**	
Suffix length	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
3	22.14	(4.05)	54.36	(3.56)	65.13	(2.33)
5	36.49	(4.94)	52.64	(1.67)	67.27	(2.21)
Complete	34.14	(7.76)	22.36	(3.71)	66.18	(3.24)

Table 6.9: Suffix of length three or five of the inflected form, for three or four left and right neighbors in addition to the name.

	±3		**±4**	
Suffix length	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
3	64.10	(2.53)	63.64	(1.87)
5	67.63	(2.55)	66.68	(2.17)
Complete	67.77	(2.69)	67.62	(2.10)

6.2.4 Window of Grammatical Category as the Only Attribute

This attribute is described in section 6.1.3, where we demonstrated this attribute on the word preceding the name: we said that two alternatives would be tried, we would either encode the lemma of the grammatical category (POS) of one reading or we would keep some of the ambiguity of the tagger by encoding the grammatical category of two readings. We also said that the respective grammatical categories of the parts of the name would not be encoded.

Table 6.10 shows results for symmetric windows of grammatical categories. The grammatical category is derived from a single reading: if there is a name reading with a name category among the readings, this is the chosen reading. The topmost reading is otherwise chosen. We will examine the alternative of encoding the POS of two readings in section 6.3.2. In the top row, the leftmost cell contains accuracy results for a classifier whose input lines contain only the shared attribute of every NE: that it is a name. The highest accuracy is reached for a symmetric window of size two, 59.24 (3.02).

The second row was included out of interest for the maximum entropy model itself. The POS of every NE is *proper name*, and we were interested in the effect of removing an attribute value, which is shared by all NEs. This is done in the second row. Comparing the results of each column, the classifier which includes the attribute that all NEs are names (top row), is consistently better than the alternative. If there are fewer attributes, for example only the grammatical category of the preceding neighbor, adding an attribute that is identical for all names does not increase accuracy: if the shared attribute of the NE is ignored, the grammatical category of its left neighbor yields an accuracy of 57.60 (3.12) versus the 57.11 (5.37) of including the POS common to every NE.

The classifier corresponding to an accuracy of 59.24 (3.02) in most cases assigns either PERSON or LOCATION. The tags assigned for a particular fold are divided as follows: 532 PERSON, 249 LOCATION and finally 21 ORGANIZATION. The three remaining categories are never assigned in this particular fold. The high frequency of PERSON can be expected as PERSON is the most common category. More interesting is the F-measure of LOCATION of at 58.09 (4.29).

Table 6.10: The grammatical category window: If there is a name reading among the readings assigned by the grammatical tagger, the POS equals name. If not, the grammatical category of the top reading is chosen. In the second row, the shared POS of all NE is ignored. The best result, 59.24 (3.02), is achieved for two left and right neighbors of the NE and the shared POS of the NE.

	NE only		**±1**		**±2**		**±3**	
	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
with NE	48.10	(5.35)	56.62	(3.47)	**59.24**	**(3.02)**	57.53	(2.08)
without NE	—	(—)	55.03	(3.90)	57.51	(3.13)	54.95	(1.92)

Table 6.11: The POS (from a single reading) of a symmetric window of two is the only attribute provided. The most interesting results are those of LOCATION and ORGANIZATION as it is not evident why a POS window is useful only for LOCATION.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Person	86.32	(1.64)	63.26	(5.25)	72.91	(3.64)
Organization	3.86	(1.99)	38.03	(15.96)	6.95	(3.40)
Location	66.41	(5.83)	52.32	(6.53)	58.09	(4.29)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall	59.24	(3.02)				

6.2.5 Results for Acronym as the Only Attribute

The acronym attribute is discussed in section 6.1.4, where we speculate if this attribute should refer to the inflected form or to a lemma. We observed instances in the corpus where only one of the two was written in block capitals: an instance of the name of the Norwegian capital, was written in capital letters in a particular newspaper text *(OSLO)*, but its lemma was not, as the name *Oslo* is found in the tagger's lexicon. An actual difference in the corpus seems very rare, as we get identical results for letting the attribute apply to the word form or lemma.

The result of only providing acronym is a model that assigns the ORGANIZATION tag to all acronyms and the PERSON tag in all other cases. This is in accordance with our intuition that it is first and foremost organization names that are acronyms. Inspection of the NE-annotated corpus showed that 299 of a total of 393 acronyms are ORGANIZATIONS. Classifying all non-acronyms as PERSON follows naturally from the fact that the PERSON tag is the most frequent category in the annotated data.

Table 6.12 shows accuracy results for the acronym attribute. Recall is high for PERSON with 99.69 (0.44), while precision is considerably lower at 50.81 (5.13), which means that practically all PERSON instances are recognized, but that in half of the instances the PERSON tag was assigned to a name of a different category. In the case of ORGANIZATION, the opposite is the case: precision is considerably higher than recall as precision is 70.29 (12.52), while recall is only 19.68 (5.61). All three mean accuracy measures of the remaining four categories are zero.

The overall performance is 52.06 (4.59), which is low in comparison with the lemma window or the grammatical category window. The reason being that barely five per cent of all names in the corpus are acronyms.[8]

6.2.6 Result for Capitalization Pattern as the Only Attribute

This attribute is described in section 6.1.4 and refers to the extent to which non-initial parts of the name are capitalized. In Norwegian, the different non-initial parts of the name can be, but are not necessarily, capitalized. We choose to distinguish between three different patterns: in the first case, all parts of the name are capitalized as in *George W. Bush*. In the second case,

[8]This result is for when the attribute applies to the inflected form of the name. If it instead applies to the lemma, results are the same: 52.12 (4.56).

Table 6.12: Acronym as the only attribute: this classifier assigns ORGANIZATION to all acronyms, PERSON to all non-acronyms. The F-measure of PERSON is two percentage points higher than in the baseline model. The low overall result can be put down to the low frequency of acronyms in the NE-corpus.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	99.69	(0.44)	50.81	(5.13)	67.18	(4.45)
Organization	19.68	(5.61)	70.29	(12.52)	30.50	(7.72)
Location	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall			52.06	(4.59)		

only the first part of the name is capitalized, as in, *Det norske vitenskapsakademi* (The Norwegian Academy of Science). Finally, there is the case where only some of the non-first parts are capitalized, as in the book title *Jødenes historie i Norge gjennom 300 år*. Multi-word names that refer to public institutions tend to have only capitalization of the initial part, contrary to names of people and companies that normally have only capitalized parts.

As for their relative frequency, the distribution of the different capitalization patterns in the NE corpus is as follows: 5 486 of all names are single word. Among the multi-word names, 1 693 of all capitalized parts, 106 with both capitalized and non-capitalized parts and finally 247 with only uncapitalized non-first parts.

The classifier that results from providing the capitalization pattern as the sole attribute resembles that of providing acronym as the only attribute, it assigns the PERSON or ORGANIZATION tag exclusively. Here, single-word names receive the PERSON tag, as well as the names whose parts are all capitalized (*George W. Bush*). Multi-word names whose non-first parts are either all non-capitalized (as in *Statens lånekasse for utdanning*) or alternatively some are (as in *Universitetet i Oslo*), are assigned the ORGANIZATION tag.

Table 6.13: The capitalization pattern as the only attribute: we record whether all, some or none of the non-first parts of a multi-word name are capitalized. This attribute is particular to Norwegian. All single word names and names with only capitalized parts are tagged PERSON, while the remaining multi-part names are tagged ORGANIZATION.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	99.49	(0.56)	50.25	(5.41)	66.62	(4.74)
Organization	12.72	(5.69)	52.93	(18.85)	20.31	(8.45)
Location	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall	50.51	(5.25)				

As with the acronym attribute (Table 6.12) recall for PERSON is much higher than precision, while the opposite is the case for ORGANIZATION, Table 6.13. The overall mean is 50.51 (5.25).[9]

There are only 26 names containing digits in the entire corpus, which explains why recording the presence of a digit does not outperform the baseline model, overall performance is 48.17 (5.35).

[9]Simplifying further, by providing the number of parts of the name as the only attribute, the overall result is at 48.57 (5.40) low.

6.2.7 List Look-up

While the effect of the acronym and capitalization attributes were much as expected, in that they do help recognize ORGANIZATION names, the results for name lists are far more surprising.

Table 6.14: List look-up: name lists are the only attributes provided. The F-measure of PERSON and LOCATION are rather close with PERSON on top, while, surprisingly, the F-measure of ORGANIZATION is close to zero.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	92.29	(21.62)	60.85	(8.56)	72.76	(13.69)
Organization	0.08	(0.26)	10.00	(31.62)	0.16	(0.51)
Location	61.63	(9.14)	78.64	(6.07)	68.83	(7.07)
Event	2.50	(7.91)	0.07	(0.21)	0.13	(0.41)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall			59.82	(11.57)		

Table 6.14 shows results for list-look up. The picture here is quite striking. The mean F-measures for PERSON and LOCATION are 72.76 and 68.83 respectively, whereas the classifier fails to classify ORGANIZATION, (the F-measure is 0.16). Hence, the situation is opposite of that for the attributes acronym and capitalization pattern, for which apart from the PERSON category, non-zero accuracy values were recorded for ORGANIZATION. The overall standard deviation is high: 11.57. For comparison, the standard deviations of the best lexical windows are between two and three. The high standard deviation probably partly indicates that the extent to which names have received a list tag varies a lot between different texts, and therefore also between the training data of the different runs.

Table 6.15 shows results limited to names that carry at least one list tag. The overall performance of this group is 15 percentage points higher than it is for all names, while the relative success for the different groups resembles that of all names: again, the F-measure of ORGANIZATION is extremely low (0.34), it is very good for PERSON and LOCATION, while it is zero for the three remaining categories.

Table 6.15: Results for names that carry at least one list tag. The mean F-measures of PERSON and LOCATION are both very similar and very high. The standard deviation of the F-measure for PERSON is with 24.39 very high.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	89.84	(27.92)	91.33	(5.20)	87.70	(24.39)
Organization	0.18	(0.55)	10.00	(31.62)	0.34	(1.09)
Location	92.55	(2.98)	78.64	(6.07)	84.92	(3.93)
Event	10.00	(31.62)	0.07	(0.21)	0.13	(0.41)
Work	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Other	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Overall			74.90	(17.61)		

The list attribute is described in section 6.1.6. Listed names total a good 13 200 names, which is a rather small number compared to the English NER system of Mikheev et al. (1999) or the Danish NER system of Bick (2004): their lists are roughly three times as big. The size of each of our lists is shown in the left column of Table 6.16: *Person* and *Location* make up the longest lists. *Location* is with close to 6 700 names the most numerous, while there are roughly 5 500 different *person* names. There are, for example, only 734 *Organization* names.

Items on a list do not necessarily all behave in the same way: in the case of the *Location* name list, names of rivers, mountains and lakes get the location list tag, whereas names that denote countries or capital cities get both the organization and location list tags. Similarly, among the names on the *Work* list, the encoded information is that the list says either ORGANIZATION or WORK, whereas a book title only gets the *Work* list tag. This distinction between items on a list was done manually. Listed person names on the other hand always results in the *Person* list tag. Multiple-part names inherit the list tags of its parts: the name *George Bush* will carry the *Person* list tag, provided *George* or *Bush* is on the list of person names.

As for the correct tag of the NEs, we explained that it is determined by function: if the country name refers to the political leadership, the correct category is ORGANIZATION. Similarly, if the newspaper name can be said

Table 6.16: The left column shows the different name types on each list: the lists of PERSON names and LOCATION names respectively are the most exhaustive. The lists for WORK, EVENT and OTHER names are much smaller, with the EVENT list being very short. The right column shows the respective number of list tags in the annotated corpus.

Category	**Number of Instances Listed**	**List tags in corpus**
Person	5 486	2 685
Location	6 690	1 706
Organization	734	884
Work	149	83
Other	138	54
Event	16	1
Total	13 213	5 413

to stand for its staff, the category is ORGANIZATION. We expect a smaller effect of our lists than if if the opposite annotation strategy had been chosen, that is one which weights form over function.

In the cross-validation data, 2 685 names were tagged the PERSON (name) list tag, 884 the ORGANIZATION list tag, 1 706 the LOCATION, only 1 the EVENT, 83 the WORK and finally 54 an OTHER tag. These numbers are shown in the right column of Table 6.16. A total of 2939 names were found on neither list. The total number of list tags exceeds (7532-2939), since a name may be assigned more than one list tag. Considering the high number of names that carried an ORGANIZATION list tag, as either its only list tag or as one of several, it surprised us that the system altogether fails to recognize ORGANIZATIONs. Markert and Nissim (2002) observe empirically that in the case of geo-political entities such as countries the two senses location and organization are too close for a successful distinction.

6.2.8 Conclusions

To sum up the effect of single attributes, we bring one plus two tables. Table 6.17 shows a sample of overall results of single attributes. We conclude that a symmetric lexical window and a suffix window come out on top. More surprisingly, the result for name lists is very similar to that of a POS window, which again illustrates the limited success of the former. Both are down 10 percentage points compared to the lemma window (top row). The results of acronym and capitalization resemble each other. The low score of 52.06 (4.59) of acronym, stems from the fact that while ORGANIZATION acronyms are assigned the correct tag, acronyms themselves only make out five percent of all names in our corpus.

Table 6.17: Overall F-measure results for single attributes in decreasing order: at the top are lemma or suffix windows followed by lists. At the bottom is the attribute that records the number of parts of the name.

Symmetric lemma window of 3 neighbors	69.65 (2.39)
Suffix length 5, window of 3	67.63 (2.55)
Name lists	59.82 (11.57)
POS window	59.24 (3.02)
Acronym	52.06 (4.59)
Capitalization pattern	50.51 (5.25)
Number of parts of the name	48.57 (5.40)
Baseline	48.10 (5.35)

Table 6.18: Results in F-measure of single attributes for PERSON, ORGANIZATION and LOCATION. We ignore the the results for the remaining categories, since they across all attributes they were near zero and with relatively large standard deviations.

	Lemma window		**Acronym**		**Pattern**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	79.60	(2.28)	67.18	(4.45)	66.62	(4.74)
Organization	58.32	(5.49)	30.50	(7.72)	20.31	(8.45)
Location	69.11	(4.92)	0.00	(0.00)	0.00	(0.00)

Table 6.19: Results in F-measure of single attributes for PERSON, ORGANIZATION and LOCATION for the remaining two attributes: a window of grammatical categories and name lists.

	POS window		**Lists**	
Category	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Person	72.91	(3.64)	72.76	(13.69)
Organization	6.95	(3.40)	0.16	(0.51)
Location	58.09	(4.29)	68.83	(7.07)

Tables 6.18 and 6.19 show (mean) F-values for three of the six categories for different single-attribute classifiers. The categories are PERSON, LOCATION and ORGANIZATION, since it is relative to these categories that the different single attribute classifiers differed the most. In the case of all three categories, the highest values were recorded for the lemma window. Table 6.18 shows that in the case of the lemma window, there is roughly a 10 percentage point difference between PERSON (on top) down to LOCATION and then again down to ORGANIZATION. For attributes other than lemma or suffix windows, PERSON constantly came out first, while either LOCATION or ORGANIZATION scored close to zero.

6.3 Results for Pairs of Attributes

We now add an additional attribute to the, by far, most important attribute, namely the lexical window. It is more natural to choose a window of lemmas than a window of suffixes. While results varied greatly when a singular attribute was provided, we expect the results of this section to be more homogeneous since the most important feature is already present.

We report results throughout this section for the different categories with the example of a lemma window of three plus one additional attribute. The smaller lemma window consisting of the name and its two left and right neighbors did have a slightly higher F-score than our the larger lemma window of three. The McNemar's test, however, showed that the differences between the results of two, three or four lemmas are not statistically significant, which justifies our choice.

We are most interested in the addition of suffixes, grammatical category and name lists. We take less interest in the addition of the attributes that record acronyms and capitalization patterns as these single-attribute classifiers were more transparent in terms of tag assignment, that is we expect less interaction between the attribute groups.

Table 6.20: Three lemmas: this is our starting point.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	82.94	(3.44)	76.48	(4.69)	79.47	(2.79)
Organization	53.70	(5.53)	62.41	(9.68)	57.42	(6.18)
Location	71.07	(4.28)	66.61	(7.59)	68.49	(4.88)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	11.00	(6.73)	19.02	(14.27)	13.20	(7.46)
Other	9.31	(11.03)	14.92	(15.36)	10.37	(11.04)
Overall			69.65	(2.39)		

6.3.1 With Suffix of Name and Neighbor

Suffixes clearly overlap, to a large extent, with the corresponding lexical window. For this reason we in section 6.2.3 only looked at how suffix windows

compared to same-size windows (of inflected forms). Table 6.21 shows results for the symmetric lemma window of three neighbors combined with different length suffixes (rows) and varying size of the suffix window (columns).

Table 6.21: The symmetric window of lemmas equals three for all experiments reported in this table. The symmetric suffix windows are of varying size combined with a varying suffix length.

	Name only	**Suf window 1**	**Suf Window 2**
n last letters	$\bar{\mathbf{X}}$ (s)	$\bar{\mathbf{X}}$ (s)	$\bar{\mathbf{X}}$ (s)
3 last letters	71.21 (2.53)	71.49 (2.75)	70.62 (2.17)
4 last letters	71.78 (2.51)	71.79 (2.41)	71.40 (2.59)
5 last letters	71.94 (2.57)	**72.25 (2.63)**	72.00 (2.57)
6 last letters	71.55 (2.70)	71.53 (2.51)	71.40 (2.47)

The values recorded for a suffix length of five are consistently highest: a suffix window of size 1 yields the highest mean 72.25 (2.63), but removing the right neighbor from this window, does not change the result: 72.26 (2.72), Table 6.22.

Table 6.22: Window reduction: the addition of a symmetric suffix window of size one (left result) is no better than using only the left neighbor and the name (while ignoring the right neighbor.) The suffix length is kept fixed at five.

Symmetric	**Left neighbor**
72.25 (2.63)	72.26 (2.72)

Table 6.23 show results for the symmetric lemma window of three neighbors and the five last letters of the preceding neighbor and the name. Compared to the lemma window by itself, the overall score is up by 2.6 percentage points.

Table 6.23: Symmetric lemma window of size three, suffix of length five of the immediately preceding word and the name itself. Compared to the lemma window of Table 6.20, the overall result is up by 2.6 percentage points, while all categories are improved somewhat.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	84.51	(3.03)	79.58	(4.48)	81.90	(2.83)
Organization	58.44	(6.17)	63.95	(9.06)	60.72	(5.75)
Location	74.24	(3.82)	68.40	(6.55)	71.01	(4.20)
Event	5.00	(15.81)	6.67	(21.08)	5.71	(18.07)
Work	15.21	(11.06)	27.70	(17.24)	18.86	(12.40)
Other	10.64	(10.19)	20.16	(13.97)	12.99	(10.61)
Overall			72.26	(2.72)		

6.3.2 With Windows of Grammatical Categories

In the previous section, we found that among symmetrical windows of grammatical categories, the window of size two was optimal. The grammatical category was derived from one reading. We also remember that the F-measure of LOCATION was just over 58, while only 7 for ORGANIZATION.

Table 6.24: Symmetric and asymmetric windows of grammatical categories.

	0 right		**1 right**		**2 right**	
	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
0 left	70.72	(2.15)	70.71	(2.30)	–	(–)
1 left	**70.95**	**(2.35)**	70.72	(2.24)	–	(–)
2 left	–	(–)	–	(–)	(70.60)	(2.20)

Table 6.24 shows the results of adding different symmetric and asymmetric windows of grammatical categories. Results for the symmetric window are shown on the diagonal: *0 left combined with 0 right* means that we are only adding the fact that the NE is a proper name. This result is identical to the one obtained from also including the POS of the left and right neighbors. Both options are better than the symmetric window of two. The highest mean (shown in bold) is achieved for the grammatical category of the preceding word and the NE. In this table, the grammatical category of a name whether the NE or a neighbor is the proper name. The POS of a non-name neighbor is, if the inflected form has been assigned more that one tag, the POS of the top ranked.

Now, if we in the case of the preceding word and the NE, instead encode the grammatical category of the top two readings assigned by the Constraint Grammar Tagger to the preceding word, if this is a non-name, results are absolutely unchanged: we again record 70.95 (2.35). Table 6.25 shows results for the different categories.

6.3.3 With Uppercase-Lowercase Attributes of the Name

Table 6.26 gives results for the different name categories for a lemma window of three and the acronym attribute. We remember that the acronym

Table 6.25: Three lemmas plus the grammatical categories of the preceding neighbor and the NE itself. The overall gain is 1.3 over the lemma window by itself. Compared to the lemma window the gain is evenly distributed among the categories.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Person	86.81	(2.39)	75.12	(4.98)	80.43	(2.59)
Organization	52.44	(5.24)	65.50	(8.89)	57.94	(5.38)
Location	70.20	(5.02)	68.32	(7.28)	68.92	(4.42)
Event	2.50	(7.91)	10.00	(31.62)	4.00	(12.65)
Work	12.36	(11.35)	25.25	(20.12)	16.13	(13.86)
Other	9.58	(8.76)	20.25	(12.66)	12.31	(9.22)
Overall			70.95	(2.35)		

alone had non-zero F-measures only for PERSON and ORGANIZATION. The mean overall F-measure is up two points relative to the lemma window alone. While PERSON and ORGANIZATION have a small improvement over the lemma window, the F-measure of LOCATION is at 68.63 (4.66) only slightly improved (by 0.14 percentage points.)

We now turn to the results for adding to the lemma window an attribute that distinguishes multi-member names into different classes depending on if all, none, or some of the non-first parts are capitalized. Table 6.27 gives the results for a lemma window with three neighbors and this attribute added. We remember that capitalization pattern by itself gave non-zero accuracy scores only for PERSON and ORGANIZATION. Compared to the lemma window by itself, the overall gain is 1.7 percentage points, while there is a small improvement for each of the three small categories. As with single attributes, we will see that acronym and capitalization again are not the better attributes.

Table 6.26: Name and three lemmas plus acronym.

	Recall		Precision		F-measure	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	86.95	(1.97)	76.07	(4.81)	81.04	(2.43)
Organization	56.60	(6.02)	68.57	(8.62)	61.77	(6.03)
Location	69.88	(4.73)	67.99	(7.42)	68.63	(4.66)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	8.36	(9.74)	18.04	(19.97)	10.99	(12.52)
Other	9.44	(10.38)	18.89	(16.92)	11.82	(11.60)
Overall			71.78	(2.43)		

Table 6.27: Name and three lemmas plus capitalization pattern.

	Recall		Precision		F-measure	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	86.40	(3.40)	77.10	(4.07)	81.37	(2.04)
Organization	53.75	(5.97)	63.59	(8.23)	57.88	(5.11)
Location	71.50	(4.14)	68.14	(8.21)	69.48	(5.13)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	15.49	(7.54)	31.69	(15.89)	20.14	(9.14)
Other	6.97	(9.50)	15.07	(16.41)	8.74	(10.05)
Overall			71.37	(1.93)		

6.3.4 With Name Lists

We examine here the final combination of a symmetric lemma window of size three with an additional attribute. To recapitulate, list-look-up by itself resulted in a very large overall standard deviation of more than 11. Moreover the F-value was good for PERSON, acceptable for LOCATION, but represented a complete failure in the case of ORGANIZATION, see page 81. Table 6.28 shows results for the lemma window and the name lists: compared to name lists alone, the leap in the F-value of ORGANIZATION (up by 62) is striking.

Table 6.28: Three lemmas plus name lists.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	88.01	(6.33)	85.01	(4.00)	86.36	(4.16)
Organization	61.27	(5.19)	65.58	(8.68)	62.83	(3.69)
Location	79.21	(4.62)	72.81	(6.48)	75.69	(4.50)
Event	3.50	(8.18)	8.33	(18.00)	4.87	(11.11)
Work	14.55	(10.22)	25.66	(15.84)	17.47	(10.33)
Other	14.95	(12.84)	21.50	(12.86)	15.39	(9.44)
Overall			75.96	(3.11)		

The result corresponding to the addition of the name lists is at close to 76 clearly better than the preceding pairs.

6.4 The Full Classifier

The different attributes are to be added one at a time until a full classifier is reached, starting with the lexical window and the name lists, the two most important groups of attributes. Table 6.29 shows the incremental addition of attributes.

Table 6.29: Starting with the symmetric lexical window of the name and three lemmas, attributes are added one at a time. Each addition is statistically significant at a significance level of 0.01. The final classifier has an overall performance of 80.15 (2.77).

	±3 **lemma**		
Row	**Attributes**	$\bar{\mathbf{X}}$	(s)
0	—	69.65	(2.39)
1	**Name lists**	75.96	(3.11)
2	**Name lists, suf5 of w-1 and w0**	77.65	(3.31)
3	**Name lists, suf5 of w-1 and w0, acronym**	79.33	(2.65)
4	**Name lists, suf5 of w-1 and w0, acronym, cappattern**	80.15	(2.77)
5	**Name lists, suf5 of w-1 and w0, acronym, cappattern, POS of w-1 and w0**	80.33	(2.57)

The suffix of length five of the preceding word form and the name is then added (row 2). The additions of the acronym and capitalization pattern attributes follow (rows 3 and 4 respectively). The recorded result increases with each attribute added, with 80.15 (2.77) as the end result. McNemar's tests yield that the addition of each attribute is significant at a significance level of 0.01.[10] Two additions to the classifier of row 4 were attempted, but not shown, as they proved not to be statistically significant: once the attribute that captures capitalization pattern is included, the addition of the number-of-parts attribute is **not** statistically significant. And, finally, if the

[10]The McNemar's test is two-sided with Yates correction. Adding suffix is statistically significant: $\chi^2 = 44.401, p = 0.00000$.
Adding acronym is statistically significant: $\chi^2 = 58.369, p = 0.00000$.
Adding capitalization pattern is significant: $\chi^2 = 17.76, p = 0.00002$.

POS of the preceding word is included, overall performance is 80.33 (2.57), but the slight increase is not statistically significant.[11] The relative success of the different NEs of the full classifier is shown in Table 6.30.

Table 6.30: The full classifier: the difference between the top-score, (PERSON) and the second-best (LOCATION) is 10, while another 10 separates number two from number three. The standard deviations of the three low-scoring categories is large even in this final model.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	92.63	(4.03)	86.04	(3.52)	89.14	(2.74)
Organization	67.09	(5.34)	72.58	(8.51)	69.41	(5.01)
Location	82.20	(2.96)	77.30	(4.89)	79.61	(3.44)
Event	2.50	(7.91)	10.00	(31.62)	4.00	(12.65)
Work	15.69	(9.04)	37.44	(22.40)	21.62	(12.19)
Other	14.13	(12.38)	26.22	(15.56)	16.81	(11.84)
Overall			80.15	(2.77)		

6.5 Parameter Optimization

The full classifier corresponding to an F-value of 80.15 (2.77) employs the default values for model-building. In this section, we examine the potential for improvement by optimizing the model building parameters.

Chapter 4 discussed alternatives to smoothing with feature frequency cut-off and alternative parameter estimation algorithms. The Maxent package does not offer alternative algorithms for feature selection and parameter estimation, which leaves the option of varying the frequency threshold and the number of iterations.

Table 6.31 shows results for different numbers of iterations of the GIS algorithm in combination with no cut-off (leftmost column) or the default cut-off threshold alternatively. No cut-off equals a frequency threshold of 1. The in-between option of a cut-off of two (results not shown), gave results between those of the default cut-off and no cut-off.

[11] The attempted final addition of POS is NOT statistically significant: $\chi^2 = 1.844, p = 0.17445$.

Table 6.31: When the option is attribute selection through frequency cut-off, no attribute selection is clearly better than a the default threshold of three.

	No cut-off, m=1	**Default, m=3**
No of iterations	$\bar{\mathbf{X}}$ (s)	$\bar{\mathbf{X}}$ (s)
50	81.27 (2.50)	80.42 (2.43)
100	**81.36 (2.57)**	80.15 (2.77)
150	81.20 (2.54)	79.77 (2.65)

The mean values $\bar{X}$ of the left column vary only a little: from 81.20 through 81.36. Furthermore, results for no cut-off (leftmost column) are consistently higher than those of the default cut-off of three. The highest recorded value is 81.36, which corresponds to 100 iterations and no cut-off, while the classifier of the previous chapter is reported in the center cell of the right column.

Statistical testing, using as before McNemar's tests with Yates correction and a significance level of 0.01, yields that results within the leftmost column are NOT statistically significant.[12] A similar McNemar's test does, however, show that the difference between the classifier of the previous chapter (100, default) and the best performing system corresponding to 81.36 (2.43), (100, no cut-off) is significant at a significance level of 0.01.[13]

Table 6.32 shows results for the classifier which has been optimized over two steps: first in terms of the attributes, then in terms of the model-building parameters. Removing the threshold means that the overall performance is up by 1 percentage point. Comparison with Table 6.30 shows that two categories actually degrade. The two are the low-frequency categories EVENT and OTHER.

[12] If, within the leftmost column, the result of the highest value and the lowest value are compared, the difference is **not** statistically significant ($\chi^2 = 4.033, p = 0.04460$).

[13] A McNemar's test concludes that the difference between the classifiers of the middle row is significant, $\chi^2 = 13.95, p = 0.00018$.

Table 6.32: The full classifier with optimal model-building parameters: the default value of 100 iterations is unchanged, but instead of a frequency threshold of three, no features of the training data are discarded. The relative success of the different categories follows the now familiar pattern.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	94.74	(2.81)	84.31	(4.45)	89.14	(2.58)
Organization	68.83	(6.17)	76.47	(6.73)	72.12	(4.04)
Location	82.66	(4.06)	78.45	(6.07)	80.32	(3.70)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	15.87	(10.94)	66.67	(40.82)	24.89	(16.47)
Other	8.00	(13.28)	27.38	(35.77)	11.30	(16.86)
Overall			81.36	(2.57)		

Chapter 7

Results Analysis

This chapter consists of two parts: in the first part, oracle accuracy and generalization capacity are recorded. Any NE-classifier is expected to do better on NEs encountered in the training data, than on NEs that are new to the system, but generalization capacity is an important feature of any system. A comparison of our result with existing results for Norwegian, Danish and Swedish constitutes the second part.

7.1 Oracle Accuracy and Generalization Capacity

In Chapter 4, we announced that we would not only measure performance by holding the most probable category against the correct category. We are also interested in to what extent the correct category is among the n most probable categories. Tables 7.1 and 7.2 together show oracle accuracy for $n = 1$ through 6. By $n = 2$, empirical means are higher than 90 for the three top results of PERSON, LOCATION and ORGANIZATION, by $n = 3$, it is more than 98.5 for the same categories. The figures representing oracle accuracy for different values of n also show the general ranking of the different categories according to the model. This means that in nearly all instances EVENT, WORK and OTHER are considered the least likely alternatives, with EVENT as the absolutely least likely alternative.

The capacity to generalize (from known to unknown) is an essential feature of any system. We therefore record separate results for known names, names in the test corpus that are also in the training corpus, and names only found in the test corpus.

The 7 532 names of the entire corpus represent 2 970 different names.

Table 7.1: Oracle accuracy is given over two tables, of which this constitutes the first. Results for $n = 1$ are of course identical to those of recall.

	1		**2**		**3**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	94.74	(2.81)	98.53	(0.91)	99.85	(0.22)
Organization	68.83	(6.17)	90.40	(3.98)	99.25	(1.06)
Location	82.66	(4.06)	93.27	(1.99)	98.65	(0.97)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	15.87	(10.94)	32.91	(20.37)	48.00	(16.03)
Other	8.00	(13.28)	14.75	(12.55)	35.86	(9.14)
Overall	81.36	(2.57)	90.85	(2.19)	95.63	(1.53)

They are different in a strict sense in that for example an extra inserted space results in a new name. Inspection shows that names that occur many times in our corpus are the names of the newspapers the texts were taken from (such as *Aftenposten*), the names of the major Norwegian cities (such as *Bergen*), and the first names of the characters of our fictional text.

An instance of the test data is known only if there is at least one identical instance in the training data. The partition into training and test data for each fold of the cross-validation was done keeping documents intact. Table 7.3 shows the respective number of known and unknown name instances in the test corpus of each fold. In half of the ten runs, known instances represent a third of all instances in the test data, in three runs, more than a quarter of the instances were known, in two runs, more than forty percent were known.

Table 7.4 shows results for known names. We immediately notice that this table records no results for the EVENT category. On page 70 we explained that if no instance of a particular tag occurred in the test corpus, we defined recall for this category to be zero. Similarly, we set precision equal to zero if the category is never assigned. If, in a run of the ten-fold cross-validation, a category neither occurs in the test data nor is ever assigned, we choose to not record results for this category in the respective run.

In the case of all names, all results were derived from ten folds. Table 7.3 confirms that the number of known names is smaller than the number of unknown names. In the case of known names, no fold records results for the EVENT category, while only eight folds record results for WORK and

Table 7.2: This is the second table which reports oracle accuracy, ie results for n equals 4, 5 or 6. Results for 6, the complete set of categories, clearly equals 100.

	4		**5**		**6**	
Category	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**	$\bar{\mathbf{X}}$	**(s)**
Person	99.98	(0.05)	100.00	(0.00)	100.00	(0.00)
Organization	99.89	(0.36)	100.00	(0.00)	100.00	(0.00)
Location	99.79	(0.28)	100.00	(0.00)	100.00	(0.00)
Event	10.75	(16.75)	23.00	(22.01)	100.00	(0.00)
Work	66.04	(15.23)	92.31	(9.22)	100.00	(0.00)
Other	81.66	(11.70)	97.86	(3.68)	100.00	(0.00)
Overall	98.10	(0.98)	99.33	(0.53)	100.00	(0.00)

OTHER.

There are a total of 39 EVENT names in the entire annotated corpus (page 52). It is therefore not surprising that no fold has an EVENT name that is common to both the training and testing data. Oracle accuracy showed that the EVENT category in most cases is judged to be the least likely alternative, so it is not surprising that it is never assigned. Furthermore, there are 145 WORK names in the annotated corpus, but only two folds lack "known" WORK names. This is probably explained by the fact that certain newspaper names are frequent in the corpus. There are 259 instances of OTHER in the entire annotated corpus, and two folds do not record results for this category.

The overall result for known names is 88.69 (2.10). The first, second and third best categories are as before PERSON, ORGANIZATION and LOCATION, but the results are clustered much closer together than is the case for all names (Table 8.4). The standard deviations of WORK and OTHER are very high.

The result for unknown names is shown in Table 7.5. In this case, 77.68 (3.06) of all names were correctly classified, which is down 11 from the known names. In the case of the three best performing categories, PERSON, LOCATION and ORGANIZATION, the difference in F-measure between known and unknown names is smallest for the best scoring category, PERSON: $(92.50 - 88.04) = 4.46$. For LOCATION the drop in performance from known

Table 7.3: The columns give the number of instances of **known** (leftmost column) and **unknown** names in the test data of each of the ten partitions. The total number of instances is given in the rightmost column.

Fold	**Nu Known instances**	**Nu Unknown instances**	**Total**
1	185	510	695
2	289	379	668
3	255	661	916
4	262	540	802
5	324	728	1052
6	134	363	497
7	313	637	950
8	273	381	654
9	200	400	600
10	242	456	698

to unknown is $(88.82 - 71.60) = 17.22$, for ORGANIZATION the drop equals 27.25. The standard deviation of ORGANIZATION for unknown names is particularly high (8.64). In the case of the WORK category, however, results are better for unknown names than for known.

We have now finished analyzing the classifier that resulted from first optimizing the attributes, then the model-building parameters: the oracle accuracy and generalization capacity were recorded. In the next section, we will compare our classifier with alternative systems for Norwegian and the highly similar languages Danish and Swedish.

Table 7.4: Results for **known** names, names that occur in the training data, for the optimized classifier.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	96.47	(4.94)	89.06	(6.47)	92.50	(4.75)
Organization	82.11	(3.16)	89.69	(5.28)	85.61	(2.59)
Location	90.71	(5.37)	87.27	(4.25)	88.82	(3.15)
Event	results not recorded					
Work	13.43	(22.32)	37.50	(51.75)	18.78	(28.80)
Other	27.78	(45.23)	30.42	(42.00)	26.82	(40.74)
Overall			88.69	(2.10)		

Table 7.5: Results for **unknown** names, names that do not occur in the training data, for the optimized classifier.

	Recall		**Precision**		**F-measure**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	94.18	(3.19)	82.87	(5.00)	88.04	(2.65)
Organization	55.89	(8.50)	62.85	(13.56)	58.36	(8.64)
Location	74.03	(5.88)	69.94	(6.36)	71.60	(3.73)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	16.49	(12.26)	69.33	(40.02)	25.73	(17.56)
Other	2.28	(3.72)	23.00	(38.89)	4.13	(6.75)
Overall			77.68	(3.06)		

7.2 Comparison

We would like to compare our result of an F-measure of 81.36 (2.57) to results achieved for Norwegian or a similar language on a similar tag set. The thesis project was part of a larger Nordic Project for proper name classification in Norwegian, Swedish and Danish. These Mainland Scandinavian languages are mutually intelligible. Different methods that included both rule-based and statistical methods were employed, while the six name categories were common to all systems. Our comparison will be with the classifiers that were developed within this project. We start by comparing our system to the two alternative classifiers for Norwegian, Nøklestad (2004) and Jónsdottir (2003). These two systems, which share name categories and annotated data with us in addition to relying on the same grammatical tagger for name detection, were described in section 2.4.

Nøklestad (2004) is a machine learning (memory) based system that predicts exactly one category for each name. Ten-fold cross-validation results on the shared annotated data for a k-value of five for all attributes, equal 81.86. (A standard deviation for results is not given). The k-value represents the number of neighbors considered by the k-nearest neighbor algorithm. So is this a significantly better result than ours? While the annotated corpus employed for training and testing is identical to ours, and in this case too, entire documents are assigned to either training or testing, the partition of each fold is not necessarily identical to ours. As for the relative success of each category, we were not able to obtain such figures. Attributes partly overlap with the ones used by us. Nøklestad (2004) reports that the memory-based system outperforms a maximum entropy classifier with identical attributes, but this maximum entropy model employs a cutoff of three, which we have found not to be optimal.[1]

Jónsdottir (2003) is a rule-based name recognizer for Norwegian, that more often than not assigns more than one category to a name. The default rule assigns all six categories to a name, while rules were written for four of the six categories, namely WORK, LOCATION, PERSON and ORGANIZATION.

Table 7.6 shows results for the Norwegian CG. The system has been tested on the annotated data, which was originally also used during rule-writing, hence the results will be somewhat higher than on an independent

[1]Nøklestad (2004) also reports results for a classifier where a normalization step is added, which in post-processing forces a name to only have one category within the borders of that document. With this normalization step, the overall performance equals 83.

Table 7.6: Results for Norwegian CG (without any post-processing step of normalization.) The consistently high figures for recall, that are dramatically higher than the precision for the same category, indicate a high level of ambiguity. From the start, it was thought that a statistical tagger could be employed to further disambiguate the tags proposed by the CG-based system.

Category	**Recall**	**Precision**
Person	98.72	61.67
Organization	85.84	33.54
Location	91.69	47.78
Event	84.62	1.24
Work	75.17	4.00
Other	87.40	8.44
Overall	93.55	32.91

test sample (Jónsdottir, 2003, page 98). These results are radically different from ours: recall is consistently high and for all categories higher than for our classifier. Unlike our system, for all categories, precision is dramatically lower than the recall of the same category. This is above all true for the three bottom categories. The precision of our classifier is for all categories better than is the case for the CG-based classifier of Table 7.6. The precision of one of the four categories for which rules were written, namely WORK, is at 4.00 very low.

The large difference between recall and precision reflects that many names receive two or more tags. We want to demonstrate how ambiguity bears upon recall and precision. Table 7.7 shows some NEs with their correct category and assigned tag(s): if we look at the recall of LOCATION, LOCATION is the correct category of three NEs (*Norway*, *Italy* and *Iceland*). All three are assigned the LOCATION tag either as the only alternative, or as one of many. Recall of LOCATION is therefore: $\frac{3}{3} \times 100 = 100$. The LOCATION tag is assigned every NE, but one (*Putin*), hence precision equals: $\frac{3}{7} \times 100 = 43$.

Results for our classifier as well as the results for the Norwegian CG and MBL systems are given in the top three rows of Table 7.8. The remaining two rows report independent large-scale results for Danish and Swedish NER systems respectively.

The Danish Constraint Grammar (CG) system is described in section 2.4.

Table 7.7: NEs with their correct tags and their tag(s) assigned by Norwegian CG.

Entity	**Correct tag**	**Selected tag(s)**
Bush	PERSON	LOCATION
Blair	PERSON	all
Newsweek	WORK	all
Norway	LOCATION	LOCATION
Italy	ORGANIZATION	ORGANIZATION, LOCATION
Italy	LOCATION	LOCATION
Putin	PERSON	PERSON, ORGANIZATION
Iceland	LOCATION	all

Like the Norwegian Constraint Grammar system it relies on a CG-based name finder. The number of name classifying rules is dramatically higher than in the case of the Norwegian system. The size of the lists employed is more than 44 000, which is more than three and a half times larger than the lists employed for the Norwegian NER. We have noted that (written) Norwegian and Danish resemble each other more strongly than Norwegian and Swedish. The Swedish NER employs context-sensitive finite-state grammars.

The results for the Norwegian ME and MBL systems are for cross-validation on a 230 000 word corpus containing some 7 500 names. Results on an independent test sample are expected to be lower than those of cross-validation, because attributes and parameters are selected and optimized relative to the latter. The Norwegian CG system was tested on a corpus of 100 000 words, which was not totally separate from the one which had been used for development. The Danish CG NER system has been tested on a Danish newspaper corpus containing 1775 names (40 000 words). The Swedish system was tested on an equally large Swedish corpus of 40 000 words, Johannessen et al. (2005).

Each system was evaluated according to the criteria used during development. The six categories employed for the Norwegian NER serve as top-nodes in the case of both the Danish and Swedish NER. The Danish system, for example, divides the LOCATION category into CIV and TOP categories, where the first, for example, applies to countries and cities, while the latter applies to names of rivers, mountains, etc. Nevertheless, what makes the Norwegian and non-Norwegian systems disparate is that in very similar semantic con-

Table 7.8: Independent large-scale evaluations of the five NER systems (Johannessen et al., 2005). All systems employ the same six name categories. While, for example, the Danish system clearly is impressive, the Norwegian and non-Norwegian systems are disparate: given very similar semantic contexts, the correct NE can differ across languages.

	Recall	**Precision**	**F-measure**
Norwegian ME	81.4	81.4	81.4
Norwegian CG	93.6	32.9	48.7
Norwegian MBL	81.9	81.9	81.9
Danish CG	95	95	95
Swedish FS	87	94	90.4

texts, the correct category is not necessarily the same across the languages. It is not uncommon that identical tag sets, for example for grammatical tagging, are applied differently in two different systems, see Johannessen et al. (2005). In the case of Norwegian NER, *function-over-form* was chosen, while the systems for Danish and Swedish chose *form-over-function*. This means that Norwegian guidelines weighted context over surface form (see sections 5.5 and 6.2.7). Examples (50) and (51) illustrate this difference: the Danish example (50) shows the correct tag for *Israel* to be LOCATION, even though it refers to the Israeli government or the Israeli Defense Forces, while in the similar Norwegian example of (51), the correct tag is ORGANIZATION.

(50) ...at Israel (LOCATION) fortsætter med at angribe de palæstinske terrorledere

...that Israel continues to attack the Palestinian terror leaders

(51) Israels (ORGANIZATION) forsvarsminister Shaul Mofaz

Israel's defense minister Shaul Mofaz

The following Norwegian example shows when a country name does actually get the LOCATION tag.

(52) en seriekamp i Danmark (LOCATION)

a league match in Denmark

The Swedish system does not distinguish between the various functions a country name can have, and uses only the LOC sense. But it does distinguish between the functions of place names and club names.

Table 7.8 shows that the Danish system outperforms the other systems, while the Swedish system is second best. The recall of the Norwegian CG-based is slightly better than that of the Danish system (96.5 to 95), but in terms of precision it is far behind, with only 38.4 to 95. The Swedish and Danish systems are clearly very successful, but we are comparing apples with oranges, when the Norwegian systems are compared with these two.

Table 7.9: The small-scale evaluations of the five NER systems (Johannessen et al., 2005). Each system is scored according to the criteria used during development.

	Nu Names	**Recall**	**Precision**	**F-measure**
Norwegian ME	115	63	63	63
Norwegian CG	115	72	52	60
Norwegian MBL	115	68	68	68
Danish CG	146	90	86	88
Swedish FS	152	91	93	91

We also performed a small-scale evaluation, reported in Johannessen et al. (2005). Since three different languages and writing conventions are involved, identical texts could not be used, but we wanted as similar test material as possible. A truly parallel corpus in the usual sense would not be desirable, since we needed original, untranslated texts. Extracts from one newspaper for each language on the same day (March 23rd 2004; *Aftenposten*, *Svenska Dagbladet* and *Politiken*) were taken. All three ran the same main stories: one about an Israeli attack on a Hamas leader, and one on the sentencing of the murderer of a Swedish politician. In addition, there were a couple stories on sports and entertainment. The corpus for each language amounted to some 1 800 tokens.

The best results are achieved by the Swedish classifier, with a score of 91 for recall, 93 for precision, see Table 7.9. The Danish CG-classifier came

close, with a recall of 90 and precision of 86. They are in another league than the Norwegian results: the MBL system achieved a recall and precision, both of 68. Our classifier only achieved 63, which is possibly an indication of over-training.[2]

In this test, our system in general predicts the correct category for PERSON. This is as expected, since cross-validation results were high for this category. The article in the small Norwegian test data on the sentencing of the murderer of a Swedish politician contained nearly exclusively the names of people, hence our result for this article was 81. The proper names in the article on sports were mainly names of sports teams which carry a location name or the names of the teams' players. The two groups are equally large. We had decided that sports teams, even when they carry location names, are to count as ORGANIZATION names. Unfortunately, our system did not assign the ORGANIZATION tag in these cases, but instead wrongly predicted the LOCATION tag. The test data of 115 proper names in the case of Norwegian, is clearly small. If, for example, the sports article had been replaced by an article on business or finance, we do not expect to reproduce our result of 63.

[2]An intermediate version of the maximum entropy classifier did poorly, with a recall and precision of 60.

Chapter 8

Conclusions

A summary of our findings constitutes the final chapter. The findings are described at length in the two previous chapters.

The classifier was optimized in two steps: first, in terms of attributes, then in terms of the model-building parameters. The attributes examined were limited to same-sentence attributes. Optimal attributes were derived in three steps: first, the result of each separate attribute was recorded. Second, the most effective attribute of the first round was combined with an additional attribute, the second attribute alternating between the remaining attributes. Finally, attributes were added one-at-a-time until a full classifier was reached. The results for attribute selection were initially reported in sections 6.2 through 6.4. The following attributes were examined: lexical windows, suffix windows and grammatical category windows. A *window* equals the NE and its neighbor(s). We also examined the effect of name lists and studied orthographic features of the name.

In the case of lexical representation, suffix and POS, both symmetric and asymmetric windows anchored at the name were examined: neighbors were removed from symmetric windows so that redundant parts were discovered. Encoding information from more than one reading was compared to deriving the attribute value only from the topmost.

Table 8.1 shows overall accuracy for single attributes (groups). The accuracy results are clustered in pairs: lemma windows and windows of suffixes came out on top at 69.7 and 67.6 respectively. Further down came name lists and windows of grammatical categories which scored 59.8 and 59.2. At the bottom came acronyms and an attribute which recorded the capitalization of name internal parts. We did expect that our annotation strategy which stresses context would make lists less effective: for example, is the correct tag of a listed location name not necessarily location. But we were surprised

Table 8.1: Overall accuracy results for single attributes in decreasing order: at the top are lemma or suffix windows followed by lists. At the bottom is the attribute that records the number of parts of the name. This table is reproduced from section 6.2.8. The baseline represents the classifier which results from the category distribution alone: the PERSON category is assigned to every name.

Attribute	$\bar{\mathbf{X}}$ (s)
Symmetric lemma window of 3 neighbors	69.65 (2.39)
Suffix length 5, window of 3	67.63 (2.55)
Name lists	59.82 (11.57)
POS window	59.24 (3.02)
Acronym	52.06 (4.59)
Capitalization pattern	50.51 (5.25)
Number of parts of the name	48.57 (5.40)
Baseline: no attributes	48.10 (5.35)

at how poorly list-look-up by itself did.

As for category-wise results for single attributes, tables 8.2 and 8.3 shows results for the categories PERSON, LOCATION and ORGANIZATION. Results for the three remaining categories (WORK, EVENT and OTHER) are not shown as they consistently were close to zero with a large standard deviation. The lemma window (leftmost column of Table 8.2) has the highest accuracy for all three categories. This column is nearly identical to results obtained for a suffix window. In all columns of the two tables, the PERSON category has the highest score among the categories. The PERSON tag is also the most common in the annotated corpus. While the lemma window has a ten percentage point difference between PERSON and LOCATION and another ten percentage points between LOCATION and ORGANIZATION, Acronym and Capitalization pattern score zero for LOCATION. Our expectations that these two attributes help recognize some organizations bear out. The relevant organization names are however not very common in the corpus: only five percent of all names are acronyms.

Table 8.3 shows that POS windows and Lists are useful for detecting LOCATION, but not ORGANIZATION. This result is more surprising than the result for acronym and capitalization pattern.

We went on to examine pairs of attributes where the most effective sin-

Table 8.2: Results in F-measure of single attributes for PERSON, ORGANIZATION and LOCATION. We ignore the the results for the remaining categories, since they were near zero across all attributes and with relatively large standard deviations.

	Lemma window		**Acronym**		**Pattern**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	79.60	(2.28)	67.18	(4.45)	66.62	(4.74)
Organization	58.32	(5.49)	30.50	(7.72)	20.31	(8.45)
Location	69.11	(4.92)	0.00	(0.00)	0.00	(0.00)

Table 8.3: Results in F-measure of single attributes for PERSON, ORGANIZATION and LOCATION for the remaining two attributes: a window of grammatical categories and name lists.

	POS window		**Lists**	
Category	$\bar{\mathbf{X}}$	(s)	$\bar{\mathbf{X}}$	(s)
Person	72.91	(3.64)	72.76	(13.69)
Organization	6.95	(3.40)	0.16	(0.51)
Location	58.09	(4.29)	68.83	(7.07)

gle attribute, a symmetrical window of the name and three neighbors, was combined with a second attribute.

Attributes were then added one-by-one. Only attributes whose addition proved statistically significant were kept. The resulting classifier has an overall performance of an F-measure equal to 80.15. This classifier employs five kinds of attributes: symmetric windows of the name and its neighbors all represented as lemmas, name lists, the five last letters of the name and its preceding neighbor, acronym and finally the Norwegian specific capitalization pattern.

Attribute Selection was achieved with the default values for the software where features occurring less than three times in the training data were ignored in terms of model-building. If instead no feature of the training data was ignored, the performance increase of a classifier employing the same attributes proved at 81.36 (2.57) to be statistically significant. When no

feature was excluded, a choice of 50, 100 or 150 iterations yielded identical results.

Table 8.4: The cross-validation results for our end classifier are achieved through attribute selection followed by parameter optimization.

	Recall		**Precision**		**F-measure**	
Category	$\bar{X}$	s	$\bar{X}$	s	$\bar{X}$	s
Person	94.74	(2.81)	84.31	(4.45)	89.14	(2.58)
Organization	68.83	(6.17)	76.47	(6.73)	72.12	(4.04)
Location	82.66	(4.06)	78.45	(6.07)	80.32	(3.70)
Event	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
Work	15.87	(10.94)	66.67	(40.82)	24.89	(16.47)
Other	8.00	(13.28)	27.38	(35.77)	11.30	(16.86)
Overall			81.36	(2.57)		

Table 8.4 shows results for our final classifier. The system does reasonably well for the three most frequent categories: PERSON, LOCATION and ORGANIZATION, while it performs poorly for EVENT, WORK and OTHER. The best results are for PERSON, then follows LOCATION and ORGANIZATION, respectively. The drop in performance is in each case around 10.

In section 7.1 generalization capacity was recorded: the performance difference between names encountered in the training data and those which are not, is found to be smaller for PERSON than for LOCATION and ORGANIZATION.

In the previous chapter our results were compared with existing name category recognition results for Norwegian and the related languages Danish and Swedish. The comparison is with one Danish and one Swedish system. Both are rule-based and use the same semantic tag set as our system, but with the difference that our tags are top-nodes in these systems. Also, what is judged the correct category varies across the systems, with the Danish and Swedish grouped against the Norwegian. The Swedish and Danish systems exploit lists more successfully than the Norwegian systems do, Johannessen et al. (2005).

Results are similar to the performance reached by a memory-based system which employs ten-fold cross validation for training and testing on the same

annotated data for Norwegian, Nøklestad (2004).

Bibliography

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.

Eckhard Bick. A Named Entity Recognizer for Danish. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 305–308, 2004.

Daniel Bikel, Scott Miller, Richard Schwartz, and Ralph Weichedel. Nymble: A High-Performance Learning Name-Finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP 1997)*, pages 194–201, 1997.

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

Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the 6th Workshop on Very Large Corpora (WVLC 1998)*, pages 152–160, 1998.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named Entity Recognition for Catalan Using Spanish Resources. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, volume I, pages 43–50, 2003.

Stanley F. Chen and Ronald Rosenfeld. A Gaussian Prior for Smoothing Maximum Entropy Models. Technical report, Carnegie Mellon University, 1999.

Hai Leong Chieu and Hwee Tou Ng. Named Entity Recognition: A Maximum Entropy Approach Using Global Information. In *Proceedings of*

the 19th International Conference on Computational Linguistics (Coling 2002), volume I, pages 190–196, 2002.

Kenneth W. Church. Empirical Estimates of Adaption: The Chance of Two Noriegas is Closer to $\frac{p}{2}$ than p^2. In *Proceedings of the 18th International Conference on Computational Linguistics (Coling 2000)*, pages 173–179, 2000.

James R. Curran and Stephen Clark. Investigating GIS and Smoothing for Maximum Entropy Taggers. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, volume 1, pages 91–98, 2003.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. Forgetting Exceptions is Harmful in Language Learning. *Machine Learning*, (34):11–43, 1999.

Hercules Dalianis and Erik Åström. Swenam—A Swedish Named Entity Recognizer. Its Construction, Training and Evaluation. Technical report, NADA–School of Computer Science and Communication, KTH–Royal Institute of Technology, 2001.

J. Darroch and D. Ratcliff. Generalized Iterative Scaling for Log-linear Models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

Thomas G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7): 1895–1924, 1998.

Ruth V. Fjeld. Simple-leksikonet. Bulletin, pages 25–27, The Text Laboratory, Faculty of Humanities, University of Oslo, April 2001.

Michael Fleischman and Eduard Hovy. Fine Grained Classification of Named Entities. In *Proceedings of the 19th International Conference on Computational Linguistics (Coling 2002)*, pages 267–273, 2002.

Tanja Gaustad. A Lemma-Based Approach to a Maximum Entropy Word Sense Disambiguation System for Dutch. In *Proceedings of the 20th International Conference on Computational Linguistics (Coling 2004)*, volume II, pages 778–784, 2004.

Joshua Goodman. Exponential Priors for Maximum Entropy Models. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Annual Meeting (NAACL 2004)*, pages 305–312, 2004.

Kristin Hagen. Automatisk sammenslåing av navn. In Henrik Holmboe, editor, *Nordisk Sprogteknologi—Nordic Language Technology 2002*, pages 351–356. Museum Tusculanums Forlag, University of Copenhagen, 2003.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, 2001.

Hideki Isozaki and Hideto Kazawa. Efficient Support Vector Classifiers for Named Entity Recognition. In *Proceedings of the 19th International Conference on Computational Linguistics (Coling 2002)*, volume 1, pages 390–396, 2002.

Andra Björk Jónsdottir. Arner, What Kind of Name is that? Master's thesis, University of Oslo, 2003.

Janne Bondi Johannessen. Named Entity Recognition in Scandinavia: The Nomen Nescio Network. In Henrik Holmboe, editor, *Nordisk Sprogteknologi—Nordic Language Technology 2003*, pages 149–157. Museum Tusculanums Forlag, University of Copenhagen, 2004.

Janne Bondi Johannessen and Paul Meurer. Automatisk gjenkjenning av vanskelige navn. In Inger Moen, Hanne Gram Simonsen, Arne Torp, and Kjell Ivar Vannebo, editors, *MONS 9*, pages 141–149. NOVUS, 2002.

Janne Bondi Johannessen, Kristin Hagen, and Anders Nøklestad. A Constraint-Based Tagger for Norwegian. In Carl-Erik Lindberg and Steffen Nordahl Lund, editors, *Proceedings of the 17th Scandinavian Conference of Linguistics*, number 19 in Odense Working Papers in Language and Communication, pages 31–48. University of Southern Denmark, Odense, 2000a.

Janne Bondi Johannessen, Kristin Hagen, and Anders Nøklestad. The Shortcomings of a Tagger. In *Proceedings of the 12th "Nordiske datalingvistikkdager" (Nodalida 1999)*, 2000b.

Janne Bondi Johannessen, Kristin Hagen, Åsne Haaland, Dimitris Kokkinakis, Paul Meurer, Eckhard Bick, Dorte Haltrup, Andra Björk Jónsdottir, and Anders Nøklestad. Named Entity Recognition for the Mainland Scandinavian Languages. *Literary and Linguistic Computing*, 20(1):91–102, 2005.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttilla. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Number 4 in Natural Language Processing. Mouton de Gruyten, 1995.

Jun´ichi Kazama. *Improving Maximum Entropy Natural Language Processing by Uncertainty-Aware Extensions and Unsupervised Learning*. PhD thesis, University of Tokyo, 2004.

Jae-Ho Kim, In-Ho Kang, and Key-Sun Choi. Unsupervised Named Entity Classification Models and their Ensembles. In *Proceedings of the 19th International Conference on Computational Linguistics (Coling 2002)*, volume 1, pages 446–452, 2002.

Dimitris Kokkinakis. Design, Implementation and Evaluation of a Named Entity Recognizer for Swedish. Technical Report GU-ISS-01-2, Språkdata, University of Gothenburg, 2001.

Robert Krovetz. More than One Sense Per Discourse. In *Proceedings of the SENSEVAL and the Lexicography Loop Workshop*, 1998.

George Lakoff and Mark Johnson. *Metaphors We Live By*. University of Chicago Press, 1980.

Robert Malouf. A Comparison of Algorithms for Maximum Entropy Parameter Estimation. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL 2002)*, pages 49–55, 2002.

Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2000.

Katja Markert and Malvina Nissim. Towards a Corpus Annotated for Metonymies: The Case of Location Names. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1385–1392, 2002.

David D. McDonald. *Internal and External Evidence in the Identification and Semantic Categorization of Proper Names*, chapter 2, pages 21–39. Language, Speech and Communication. The MIT Press, 1996.

Andrei Mikheev, Marc Moens, and Claire Groover. Description of the LTG System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 1998.

Andrei Mikheev, Marc Moens, and Claire Groover. Named Entity Recognition without Gazetteers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, pages 1–8, 1999.

Cheng Niu, Wei Li, and Rohini K. Srihari. Weakly Supervised Learning for Cross-Document Name Disambiguation Supported by Information Extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 597–605, 2004.

Anders Nøklestad. Memory-Based Classification of Proper Names in Norwegian. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 439–442, 2004.

Bengt Pamp. Övriga namn och andra; ett forsök till gruppering av egennamnen. In Kristinn Jóhannesson, Hugo Karlsson, and Bo Ralph, editors, *Övriga namn*, number 56 in NORNA-rapporter, pages 49–62. Norna-Förlaget, Uppsala, 1994.

John A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 2nd edition, 1995.

Ellen Røyneberg. AIDaS—Automatisk identifisering av stedsnavn i nyhetstekster. Master's thesis, NTNU–The Norwegian University of Science and Technology, 2005.

Erik F. Tjong Kim Sang. In *Memory-Based Named Entity Recognition*, pages 1–4, 2002. Workshop Coling 2002.

Satoshi Sekine and Chikashi Nobata. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1977–1980, 2004.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. A Decision Tree Method for Finding and Classifying Names in Japanese Texts. In *Proceedings of the 6th Workshop on Very Large Corpora (WVLC 1998)*, 1998.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-Criteria-Based Active Learning for Named Entity Recognition. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 590–597, 2004.

Thamar Solorio. Exploiting Named Entity Taggers in a Second Language. In *ACL-05 Companion Volume to the Proceedings of the Main Conference. Proceedings of the Student Research Workshop*, pages 25–30. ACL, 2005.

Erik Velldal. Modeling Word Senses with Fuzzy Clustering. Master's thesis, University of Oslo, 2003.

Finn-Erik Vinje. *Skriveregler, Bokmål, Gjennomgått av Norsk språkråd og anbefalt for offentlig bruk av Kultur- og kirkedepartementet.* Aschehoug, 8 edition, 2004.

GuoDong Zhou and Jian Su. Named Entity Recognition Using an HMM-Based Chunk Tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 473–480, 2002.

Appendix A

The Texts of the Corpus

Table A.1: The NE-annotated corpus comprises 230 000 tokens. Between a forth and a fifth of the corpus is fiction, while newspapers and magazines make up the rest. The contemporary fiction constitutes 10 excerpts, of approximately 5000 words each. This table gives the name of the authors and the title.

	Author	**Title**	**Nu of words**
1	Alnæs, Karsten	Gaia	5 646
2	Bjørnstad, Kjetil	Vinterby	5 038
3	Carling, Finn	Under aftenhimmel	5 017
4	Christensen, Lars Saabye	Hermann	5 018
5	Faldbakken,Knut	Adams dagbok	4 883
6	Haslund, Ebba	Det hendte ingenting	5 028
7	Hauger, Torill Thorstad	Røvet av vikinger	5 020
8	Lie, Sissel	Løvens hjerte	5 065
9	Staalesen, Gunnar	I mørket er alle ulver grå	5 174
10	Vik, Bjørg	Kvinneakvariet	5 297
	Total		**51 186**

Table A.2: Thirteen newspapers and magazines are represented in the NE-annotated corpus. The largest extracts are from Aftenposten, Adresseavisen and Familien with 38 116, 31 434 and 25 597 tokens respectively. Eight media are represented with between 5 000 and 8 000 words. *The figures for Dagbladet and Bergens Tidende are approximate values.

		Nu of words
1	Aftenposten	38 116
2	Dagbladet	5 000*
3	Verdens Gang	5 024
4	Adresseavisen	31 434
5	Stavanger Aftenblad	5 146
6	Vårt Land	5 680
7	Bergens Tidende	23 000*
8	Bondebladet	4 124
9	Universitas	5 232
10	Det Nye	6 163
11	Henne	18 531
12	Familien	25 597
13	Motor	7 905