UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Aleksandr Tkachenko

Named Entity Recognition for the Estonian Language

Master's thesis (20 cp)

Supervisor: Konstantin Tretyakov, M.Sc.

Autor: .... "....." mai 2010
Juhendaja: .... "....." mai 2010

TARTU 2010

Contents

Introduction

Nowadays a huge amount of the world's information is stored in the form of text in natural language, contained in Web pages, news articles, research papers, e-mails and blogs. While these text documents can be effectively searched and ranked by modern search engines, more demanding analytical tasks such as data mining and decision support require much more detailed and fine-grained processing. Knowledge confined within natural language can be made more accessible for machine processing by means of transforming the text into structured, normalised database form. Information Extraction aims to do just this – its goal is to automatically extract structured information from unstructured text documents using natural language processing. One basic sub-task in Information Extraction involves the recognition of predefined information units such as names of persons, organisations, locations, and numeric expressions including time, date, money and percent expressions. Named Entity Recognition (NER) is the process of identifying these entities in text.

While the problem of NER has been extensively studied for widely spoken languages with the state-of-the-art systems achieving near-human performance, no research has yet been done in regards to Estonian so far.

In this thesis we study the applicability of recent statistical methods to extraction of named entities from Estonian texts. In particular, we explore two fundamental design challenges: choice of inference algorithm and text representation. We compare two state-of-the-art supervised learning methods, Linear Chain Conditional Random Fields (CRF) and Maximum Entropy Model (MaxEnt). In representing named entities, we consider three sources of information: 1) local features, which are based on the word itself, 2) global features extracted from other occurrences of the same word in the whole document and 3) external knowledge represented by lists of entities extracted from the Web. To train and evaluate our NER systems, we assembled a text corpus of Estonian newspaper articles in which we manually annotated names of locations, persons, organisations and facilities. In the process of comparing several solutions we achieved F_1 score of 0.86 by the CRF system using combination of local and global features and external knowledge.

This thesis is organised as follows. In Chapter 1 we formulate the task of named entity recognition, discuss main challenges in the field, give a brief overview of the techniques that were proposed for addressing the NER problem and describe common forms of representation of named entities. We present supervised learning algorithms that we used in Chapter 2. In Chapter 3 we describe the sources of data that were used, the preprocessing steps we performed on the data and the method that was used for system evaluation. Finally, we illustrate experiments and results in Chapter 4.

Chapter 1

Named Entity Recognition (NER)

In this chapter we formulate the task of named entity recognition, discuss main challenges in the field, give a brief overview of the techniques that were proposed for addressing the NER problem and describe common forms of representation of named entities.

1.1 The Task

The task of named entity recognition refers to the extraction of atomic elements in texts and classifying them into a set of predefined categories of interest. The term “Named Entity” was first introduced in 1996 at the Sixth Message Understanding Conference (MUC-6) [GS96], focused on extracting information of company activities. In defining the task, people noticed that it was essential to recognise information units such as names of persons, organisations, locations, and numeric expressions such as time, date, money and percent expressions. Since then, NER has been recognised as an important preliminary step in many natural language processing applications, such as text summarisation, information filtering, relation extraction and question answering.

More specifically, NER can be treated as a problem of mapping a sequence of words in a block of text to a sequence of corresponding categories. A typical NER system takes as an input a chunk of text, such as

“Belgian international Luc Nilis scored twice on Sunday as PSV Eindhoven came from behind to beat Groningen 4-1 in Eindhoven.”,

and outputs a sequence of name-tagged words, such as

“O(Belgian) O(international) PER(Luc Nilis) O(scored) O(twice) O(on) O(Sunday) O(as) ORG(PSV Eindhoven) O(came) O(from) O(behind) O(to) O(beat) ORG(Groningen) O(4-1) O(in) LOC(Eindhoven).”, where the label PER refers to person names, LOC – location names, ORG – organisation names and O denotes words falling outside the set of defined categories.

1.2 What Is a Named Entity?

A named entity is a word or a phrase that refers to a particular kind of object in the world. The string *John Smith*, for instance, refers to a particular person and is therefore a named entity of type *person*. Analogously, the string *Ford Motor Company* refers to the automotive company created by Henry Ford in 1903 and is therefore a named entity of type *company*. In earlier works, NER was formulated as a problem of extracting "proper names" only [CS92, Thi95]. However, as it was recognised that for practical needs it would be beneficial to extract also other types of entities, the definition of the task was loosened to include some common names, such as temporal expressions, measures, names of biological species and substances.

The most widely studied types of entities are names of *persons*, *locations* and *organisations*. These basic types can be further subdivided into more specific categories. For instance, the type *location* can be split into sub-types such as city, state, country, etc. [Fle01, LGL05]. The fine-grained categories of the type *person*, such as *politician* and *entertainer* appear in the work of Fleischman [FH02]. The ACE program [ACE] defines the type *facility* which subsumes entities of the type *location* and *organisation*. The type *GPE* (Geo-Political Entity) is used to represent locations which have governments, such as a city or a country. MUC-6 introduced temporal expressions, such as *date* and *time*, and numerical expressions, such as *money* and *percent*. Finally, new types are sometimes defined for specific needs: *film*, *phone number*, *email address*, *book title*, *job name* [ZWB^{+}99, Bri98, CS04].

An effort has been done to create more elaborate hierarchies of name types. Sekine's hierarchy, proposed in 2002, defines about 200 categories covering the most frequent name types appearing in newspaper articles [SN]. It contains many fine grained subcategories, such as *museum*, *river* or *airport*, and adds a wide range of categories, such as *product* and *event*, as well as *substance*, *animal*, *religion* or *color*. BBN hierarchy, assembled for the Question Answering task [BBN], defines 29 types and 64 subtypes.

With the emerging interest in bioinformatics, many studies have been dedicated to extraction of names of genes, proteins, cell lines and cell types [SZZ^{+}03, Set04, TT03]. Related work also includes names of drugs and chemicals [RTWH00, NRVsAs03].

1.3 Challenges

Despite the fact that the definitions of name categories are quite clear, very often one string can represent several entity types, depending on the context. For instance, the word *Heathrow* in the context *she met him at Heathrow* is a *location*, but in the context *the Heathrow authorities* refers to an *organisation*. This phenomenon is known as metonymy. A simplistic solution is just to disregard metonymous uses of words. In this case *Heathrow*, for instance, will be always labeled as an *organisation*. However, this approach may not be very useful for practical applications of NER (e.g. in a sports domain). Idealistic solutions, on the other hand, are not always practical to implement.

Many models depend on local information to classify named entities. This can be troublesome when neither the entity string nor its context provide positive evidence of the correct entity type. In such cases, the task may be difficult even for a human, if he has no prior knowledge about the entity. Consider the sentence *Phillip Morris announced today that...* The verb *announced* is used frequently following both people and organisations, therefore contextual clue does not help to disambiguate the entity type. *Phillip Morris* actually looks a lot like a person name, and without a gazetteer of names, it would be impossible to know that *Phillip Morris* is in this case a company.

Another major issue is language non-regularity over different textual genres (journalistic, scientific, informal, etc.) and domains (sports, business, etc.). The style of a text may be influenced by a number of factors, such as form of media (e.g. emails, transcribed spoken text, written text, web pages), text type (e.g. reports, letters, books, lists), degree of formality and author. For example, less formal texts may not follow standard capitalisation, punctuation or even spelling formats. A few studies devoted specifically to different genres and domains have clearly demonstrated that although any domain can be reasonably supported, porting a system to a new domain or textual genre remains a major challenge [MTU^{+}01, MWC05].

1.4 Learning Methods

For a system to be able to handle named entities in various contexts, it is essential to identify key extraction and classification rules. The earliest works on named-entity recognition involved using hand-crafted rules. For instance, a sequence of capitalised words preceded by *Mr.* is typically the name of a person, so one could represent this observation as a rule. However, this approach might require months of work by experienced computational linguists to achieve good performance.

The current dominant technique for addressing the NER problem is supervised learning [TKSDM03]. Typical supervised learning systems induce disambiguation rules automatically by identifying discriminative features of different types of named entities in a collection of training examples. For instance, a system might learn that 95% out of all examples followed by *Inc.* are labeled as *organisation*. This observation (possibly, in combination with many others) can be then used to recognise organisations quite accurately. The downside of supervised methods is the need for a large, manually annotated training corpus.

Frequently used supervised learning techniques include Hidden Markov Models (HMM) [BMSW97], Maximum Entropy Models (MaxEnt) [BSAG98], and Linear Chain Conditional Random Fields (CRF) [ML03a].

MaxEnt is capable of utilising an extraordinarily diverse range of knowledge sources in making its tagging decisions. These knowledge sources include information about the word's capitalisation, its neighboring words, its prefixes and suffixes, its membership in predetermined lists of people and locations, and so on. MaxEnt has been shown to successfully handle millions of such features [VS07, DFM$^+$04]. A technique employed by MaxEnt is to classify each word independently. The problem with this approach is that it assumes that given a sequence of words, all of the named entity labels are independent. In fact, the named entity labels of neighboring words are dependent. For example, while *New York* is a location, *New York Times* is an organisation.

This independence assumption can be relaxed by arranging the class variables in a linear chain. This is the approach taken by the hidden Markov model (HMM) [Rab89]. In this case, any local decision depends on prediction at a previous position. However, HMMs have one significant shortcoming. Unlike MaxEnt, HMM relies on only one feature, the word's identity. But many words, especially proper names, will not have occurred in the training set, so the word-identity feature is uninformative. To label unseen words, we would like to exploit other features of a word. Enhancing HMM to handle such interdependent features is difficult to do while retaining tractability. Other option implies doing unrealistic independence assumptions among the features.

CRFs combine the benefits of both MaxEnt and HMM. They can be viewed as a sequential extension of MaxEnt. On the one hand, CRFs make similar assumptions on the dependencies among the class variables as HMM and, on the other hand, allow to use a rich set of features like MaxEnt.

Other methods mentioned in the literature include Support Vector Machines [AM03], Decision Trees [Sek98] and the Perceptron [Col02a].

Current NER systems often do not rely only on a single technique, but combine outputs of multiple taggers. Whenever more than one tagger is used in parallel, post-processing must be done to resolve conflicting results and make a final decision. This can be implemented by using meta-learning, the simplest form of which are voting schemes. The system in [Zho04] uses an ensemble of two HMMs and one SVM classifier with majority voting. The two HMMs are trained on different corpora. This seems to enable the system to properly adapt to different corpus properties. Another meta-learner is described in [MR04]. Here, three different SVMs are used that are trained on different corpora using different feature sets. A fourth SVM takes the results of the three original SVMs as features and generates the final result.

1.5 Feature Space for NER

Features are characteristic attributes of words designed for algorithmic processing. An example of a feature is a Boolean variable with the value true if a word is capitalised and false otherwise. Typically, a word can be characterised by a set of Boolean, nominal and numeric attributes. For instance, a NER system might represent each word with 3 attributes:

1. a nominal attribute corresponding to the lowercased version of the word,
2. a Boolean attribute with the value *true* if the word is capitalised and *false* otherwise,
3. a numeric attribute corresponding to the length of the word.

Then the text *The University of Tartu* would be represented in a form

<the, true, 3>, <university, true, 10>, <of, false, 2>, <tartu, true, 5>

Typically, the problem of NER is approached by applying a rule system over the features. For instance, a system might have two rules, a recognition rule: *capitalised words are candidate entities* and a classification rule: *the type of candidate entities of length greater than 3 words is organisation.* In fact, real systems tend to be much more complex and their rules are often induced by automatic learning techniques.

In this section, we present the features most often used for the recognition of named entities.

1.5.1 Local Features

Local features are based on information derived from the character makeup of words. Table 1.1 lists frequently used local features. These features in isolation have been successfully applied to recognising organisation, person and location names, times, dates, percentages and monetary amounts [BMSW97].

Group	**Feature**
Lexicon	- Word itself (e.g., Mustamäel) - Word in a normalised form (e.g. mustamägi)
Case	- Starts with a capital letter - Word is first in a sentence - Word is all uppercased - Word is mixed case (e.g., ProSys, eBay)
Punctuation	- Word is a punctuation mark - Contains internal apostrophe, hyphen or ampersand - Ends with period, has internal period (e.g., St., I.B.M.)
Part-of-speech	- proper name, verb, noun, foreign word
Number	- Word is a digit - Word is a Roman number - Word contains digits (e.g., W3C, 3M)

Table 1.1: Features based on the token string.

Morphological features are related to words affixes and roots. For instance, a system may learn that locations often end in *maa* (Harjumaa, Saksamaa) or that organisations often end in *amet* and *liit* (Maksuamet, Euroliit).

Word pattern features, introduced by Collins [Col02b], are designed to map words onto a small set of patterns over character types. For instance, a pattern feature might map all uppercase letters to “A”, all lowercase letters to “a”, all digits to “0” and all punctuation to “-”. In this case a word *Tarbija24.ee* would be represented as *Aaaaaaa00-aa*. The summarised pattern feature is a form of the above in which consecutive character types are not repeated in the mapped string. For instance, the preceding example takes the form *Aa0-a*.

1.5.2 External Resources

Gazetteers The terms *lexicon*, *list* and *dictionary* are often used interchangeably with the term *gazetteer*. Including list as a feature is a way to express the relation *is a* (e.g., *Tallinn is a city*). It may seem obvious that if a word (*Tallinn*) is an element of a list of cities, then the probability of this word to denote a city in a given text is high. However, because of word polysemy and metonymy, the probability is almost never 1. (e.g., in the context, *. . . klubis Tallinn toimus . . .* , *Tallinn* refers to an organisation). It turns out that the injection of gazetteer matches as features is critical for good performance of NER systems [CS04, KT07, TM, FIJZ03].

Several types of gazetteers are mentioned the literature. Most frequently lists of entities are used. These contain organisations, person first and last names, geographical locations, astronomical bodies, etc. Lists of common nouns were applied, for instance, to disambiguate capitalised words in ambiguous positions, such as sentence beginning [Mik99]. Many authors propose to recognise organisations by identifying words that are frequently used in their names [Mcd96, KGW^{+}95]. For instance, knowing that *ehitus* is frequently used in organisation names could lead to the recognition of *Merko Ehitus* and *Facio Ehitus*.

Several approaches have been proposed to automatically extract comprehensive gazetteers from the web and from large collections of unlabeled text [ECD^{+}05, RJ99]. Recently, Toral and Munoz [TM] have successfully constructed high quality and high coverage gazetteers from Wikipedia.

Unlabeled Text While labeled data is expensive to obtain, unlabeled data is often available "for free" in large quantities.

Usability of unannotated data was extensively studied in the CoNLL-2003 shared task [TKSDM03]. Participating systems used unannotated data for extracting training instances [BONV03, HvdB03] or obtaining extra named entities for gazetteers [MD03, ML03b]. A number of systems employed unannotated data for obtaining capitalisation features for words.

Word clusters generated from unlabeled data have been successfully adapted by many systems [RR09, Lia05, MGZ04]. This technique, pioneered by [BdM^{+}92], hierarchically clusters words based on their co-occurrence statistics in a large corpus. For example, since the words Friday and Tuesday often appear in similar contexts, the algorithm will assign them to the same cluster. Within a binary tree produced by the algorithm, each word can be uniquely identified by its path from the root. Using path prefixes of different length as features allows to provide different levels of word abstraction. This technique allows to alleviate the data sparsity problem common in NLP tasks. Consider, for instance, the sentence fragment *Microsofti asutaja Bill Gates.* A NER system might find it troublesome to classify the entity *Bill Gates* if neither *Bill Gates* nor *asutaja* were previously observed in a training data. However, the word *asutaja* might fall into the same cluster with the words *direktor*, *juhataja* and *professor*, which are likely to have been recognised as good predictors of the type *person*.

1.5.3 Global Features

Context from the whole document can be important in classifying a named entity. Consider, for instance, the sentence *McCann initiated a new global system.* Here, *McCann* can be a person or an organisation. Observing further in the text the fragment *CEO of McCann* can help to disambiguate *McCann* as the type *organisation.*

Chieu and Ng [CN02] identify multiple occurrences of the same word in a document and aggregate the context the word appears in. They check, for instance, whether a word is present in a capitalised form in an unambiguous position. Other features are: *the longest capitilised sequence of words in the document which contains the current word* and *the token appears before a company marker such as ltd, elsewhere the in text.*

Ratinov and Roth [RR09] observed that named entities in the beginning of documents tend to be more easily identifiable and match gazetteers more often. This is due to the fact that when a named entity is introduced for the first time in text, its canonical name is used, while in the following discussion abbreviated mentions, pronouns, and other references are used instead. To exploit this fact, they record the label assignment distribution for all token instances for the same token type in the document and use this information as features.

Chapter 2

Machine Learning Background

In this chapter we give overview of two supervised machine learning methods: Maximum Entropy Model (MaxEnt) and Linear Chain Conditional Random Fields (CRF). First, we formally define the problem of a named entity recognition. Second, we describe methods to integrate word features into a learning framework. Third, we derive a Maximum Entropy Model from the perspective of the Principle of Maximum Entropy. Finally, we present Linear Chain Conditional Random Fields.

2.1 Problem Formulation

We treat the task of named entity recognition as a text sequence tagging problem. The objective can be described as follows. Given a sequence of observations $\vec{x} = (x_1, \ldots, x_n)$ and a predefined set of class labels Y find the sequence of class labels $\vec{y} = (y_1, \ldots, y_n) \in Y^n$ with the highest conditional probability among all possible label sequences:

$$\vec{y}^* = \underset{\vec{y}}{\operatorname{argmax}}\, p(\vec{y}|\vec{x}).$$

In the following, we describe two different approaches to estimate the conditional probability of a label sequence.

2.2 Feature Functions

Feature functions are the key components of both MaxEnt and CRF. Two types of feature functions are used: state function and transition function. *State feature function* $s(y_j, \vec{x}, j)$ is a function of the label at position j and the observation sequence; *transition feature function* $t(y_{j-1}, y_j, \vec{x}, j)$ is a function of the entire observation sequence and the labels at positions j and $j-1$ in the label sequence.

When defining the feature functions, we construct a set of real-valued functions $b(\vec{x}, j)$ of the observation sequence to expresses some characteristic of the training data. An example of such a function is

$$b(\vec{x}, j) = \begin{cases} 1, & \text{if the observation at position } j \text{ is the word “Tallinn”} \\ 0, & \text{otherwise.} \end{cases}$$

Each feature function takes on the value of one of these real-valued functions $b(\vec{x}, j)$ if the current state (in the case of a state function) or previous and current states (in the case of a transition function) take particular values. All feature functions are therefore real-valued. For example, consider the following state function:

$$s(y_j, \vec{x}, j) = \begin{cases} b(\vec{x}, j), & \text{if } y_j = \text{ LOCATION} \\ 0, & \text{otherwise.} \end{cases}$$

An analogous transition function takes the form:

$$t(y_{j-1}, y_j, \vec{x}, j) = \begin{cases} b(\vec{x}, j), & \text{if } y_{j-1} = \text{ OTHER and } y_j = \text{ LOCATION} \\ 0, & \text{otherwise.} \end{cases}$$

The models assign an individual weight λ to each feature function f. These weights are learned from the training data. If $\lambda > 0$, whenever function f is active (i.e., we see the word Tallinn at a current position and we assign it label LOCATION), it increases the probability of the label sequence $\vec{y}$. This is another way of saying the model should prefer the label LOCATION for the word Tallinn. If on the other hand $\lambda < 0$, the model will try to avoid the label LOCATION for Tallinn.

2.3 Maximum Entropy Model

With the Maximum Entropy Model, instead of directly computing probability $p(\vec{y}|\vec{x})$, we decompose the problem into a set of local predictions $p(y_j|x_j)$ at each position in an input sequence j, and then combine these predictions into the final solution.

Local Prediction Let y denote the output label and x contextual information of a word at some position j in a sequence $(\vec{y}, \vec{x})$. MaxEnt is a method for estimating the conditional probability $p(y|x)$ that, given context x, the process will output y. This approach is based on the Principle of Maximum Entropy [Jay57] which states that if incomplete information about a probability distribution is available, the only unbiased decision is to select the most uniform distribution given the available information. A mathematical measure of uniformity of a distribution is entropy. In the case a conditional distribution $p(y|x)$ we use the notion of conditional entropy $H(y|x)$, defined as

$$H(y|x) = -\sum_{(x,y)\in Z} p(y,x)\log p(y|x).$$

Here the set $Z = X \times Y$ consists of X, the set of all possible input variables x, and Y, the set of all possible output variables y. The entropy is bounded from below by zero, the entropy of a model with no uncertainty at all, and from above by $\log |Y|$, the entropy of a uniform distribution over all possible $|Y|$ values of y.

The Principle of Maximum Entropy suggests to use a model $p^*(y|x)$ which, on the one hand, has the highest possible conditional entropy and on the other hand is consistent with the evidence from the training data:

$$p^*(y|x) = \operatorname*{argmax}_{p(y|x)\in P} H(y|x), \tag{2.1}$$

where P is the set of all models consistent with the training material.

We represent the training material in terms of useful facts it contains. An example of such a fact is that a frequency with which *Tallinn* is labeled as *location* is 0.1. The facts are expressed by means of stationary feature functions $s_i(x,y) \in \{0,1\}$ $(1 \leq i \leq m)$ which depend on both the input variable x and the class variable y (for simplification we omit here dependence on the input sequence $\vec{x}$). Frequency of each such fact can then be expressed as the expected value of the corresponding stationary feature function s with respect to the empirical distribution $\tilde{p}(x,y)$ in the training data:

$$\tilde{E}(s) = \sum_{(x,y)\in Z} \tilde{p}(x,y)s(x,y).$$

Now that we have formulated important statistical facts inherent to the training sample, we require that our model of the process also accords with them. We do it by constraining the expected value that the model assigns to the corresponding feature s. The expected value of a feature s on the model distribution is

$$E(s) = \sum_{(x,y)\in Z} \tilde{p}(x)p(y|x)s(x,y),$$

where $\tilde{p}(x)$ is the empirical distribution of x in the training sample.

The expected value of each feature function s_i on the particular model distribution is constrained to be the same as its expected value on the empirical distribution:

$$E(s_i) = \tilde{E}(s_i) \ (1 \leq i \leq m)$$

Equation (2.1) can then be rewritten as a constrained optimisation problem

$$p^*(y|x) = \operatorname*{argmax}_{p(y|x)} H(y|x) \quad \text{subject to } E(s_i) = \tilde{E}(s_i) \quad \text{for all } (1 \leq i \leq m).$$

It can be shown [KTK07] that the optimal solution takes the form

$$p^*(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{m} \lambda_i s_i(x,y)\right), \tag{2.2}$$

where $Z(x)$ is a normalisation function defined as

$$Z(x) = \sum_{y \in Y} \exp\left(\sum_{i=1}^{m} \lambda_i s_i(x,y)\right).$$

Maximum Entropy Model for Sequence Tagging Expression 2.2 provides prediction for a single word. In the context of sequence tagging we also need to track word's position index. For this purpose we rewrite equation 2.2 as

$$p(y_j|\vec{x}, j) = \frac{1}{Z(\vec{x}, j)} \exp\left(\sum_{i=1}^{m} \lambda_i s_i(y_j, \vec{x}, j)\right),$$

where j is a position indicator of a word in a sequence. The conditional probability of a label sequence can then be formulated as

$$p(\vec{y}|\vec{x}) = \prod_{j=1}^{n} p(y_j|\vec{x}, j). \tag{2.3}$$

2.4 Linear Chain Conditional Random Fields

Linear Chain Conditional Random Fields can be considered as a sequence version of Maximum Entropy Models. They allow to directly model the conditional probability $p(\vec{y}|\vec{x})$ without a need for factorisation 2.3. A Linear Chain Conditional Random Field defines the conditional probability

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{j=1}^{n} \psi_j(\vec{x}, \vec{y}),$$

where ψ_j are potential functions each in the form

$$\psi_j(\vec{x}, \vec{y}) = \exp\left(\sum_{i=1} \lambda_i t_i(y_{j-1}, y_j, \vec{x}, j) + \sum_{k=1} \mu_k s_k(y_j, \vec{x}, j)\right).$$

Here $t_i(y_{j-1}, y_j, \vec{x}, j)$ is a transition feature function, $s_k(y_j, \vec{x}, j)$ is a state feature function and λ_i and μ_k are parameters to be estimated from training data.

The normalisation to interval $[0, 1]$ is given by

$$Z(\vec{x}) = \sum_{\vec{y} \in Y^n} \prod_{j=1}^{n} \psi_j(\vec{x}, \vec{y}).$$

Summation over Y^n, the set of all possible label sequences, is performed to get a feasible probability.

Chapter 3

Data and Evaluation

In this chapter we describe the sources of data that were used, the preprocessing steps we performed on the data, the format of the data and the method that was used for system evaluation.

3.1 Data

To train and evaluate our NER systems, we assembled a text corpus of Estonian newspaper articles. The corpus consists of 496 articles published in the local online newspaper *Delfi* in the category "daily news" over a time period between year 1997 and 2009. The total size of the corpus is 84175 tokens.

3.2 Data Preprocessing

The raw data was preprocessed using the tool *t3mesta* [HJK98]. The processing steps involve tokenisation, part-of-speech tagging, grammatical and morphological analysis. The resulting dataset was then manually name-tagged using the GATE editor [GAT]. We distinguish four types of entities: names of *persons*, *locations*, *organisations* and *facilities*. Words that do not fall into any of these categories are tagged as *other*. The name categories are defined as follows:

Person entities refer to named persons, families or certain designated non-human individuals, such as fictional characters and named animals. Examples of such entities are: Toomas Hendrik Ilves, Sherlock Holmes, Batman.

Location entities are names of politically or geographically defined locations, such as cities, provinces, countries, international regions, bodies of water, mountains and astronomical bodies. These include for example, Eesti vabariik, Harjumaa, Haabersti, Euroopa, Munamägi and Kuu.

Organisation entities designate named governmental, commercial, educational, entertainment or other organisational structures. Examples of such entities are:

Euroopa Liit, Philip Morris, Tallinna Saksa Gümnaasium, ETV, BrainStorm.
Facility entities are limited to functional, primarily man-made structures. These include buildings and similar facilities designed for human habitation, such as houses, factories, stadiums, office buildings, gymnasiums, prisons, museums; elements of transportation infrastructure, including streets, highways, airports, ports, train stations, bridges, and tunnels. Roughly speaking, facilities are artifacts falling under the domains of architecture and civil engineering. Examples of facility entities are: Estonia pst., Tammsaare park, Locarno lennujaam, Tallinna Pühavaimu kirik, klubi Atlantis.

3.3 Data Format

The created data files contain one word per line with empty lines representing sentence boundaries. At the end of each line there is a tag which states whether the current word is inside a named entity or not. The tag also encodes the type of named entity. Here is an example sentence:

11.	11.+0	_N_ ord ? digit	O
juunil	juuni+l	_S_ com sg ad	O
laastas	laasta+s	_V_ main indic impf ps3 sg ps af	O
tromb	tromb+0	_S_ com sg nom	O
Raplamaal	Rapla_maa+l	_S_ prop sg ad	LOC
Lõpemetsa	Lõpe_metsa+0	_S_ prop sg gen	B-LOC
küla	küla+0	_S_ com sg part	LOC
.	.	_Z_ Fst	O

Each line contains four fields: the word, its lemma, its grammatical attributes [MSC] and its named entity tag. Words tagged with O are outside of named entities. Whenever two entities of a type XXX are immediately next to each other, the first word of the second entity will be tagged as B-XXX in order to show that it starts another entity. The data contains entities of four types: *persons* (PER), *organisations* (ORG), *locations* (LOC), *facilities* (FAC). We assume that named entities are non-recursive and non-overlapping. When a named entity is embedded in another named entity, usually only the top level entity has been annotated.

Table 3.1 illustrates the number of named entities in the corpus.

PER	LOC	ORG	FAC	Total
2547	2899	2098	252	7796

Table 3.1: Number of named entities in the corpus.

3.4 Evaluation

To evaluate the performance of our methods, we use two standard measures: precision and recall. Precision is the percentage of named entities found by the system that are correct. Recall is the percentage of named entities present in the corpus that are found by the system. Formally

$$\text{Precision} = \frac{|T \cap M|}{|M|} \qquad \text{Recall} = \frac{|T \cap M|}{|T|},$$

where T is a set of named entities in a corpus and M is the set of named entities that the method reports. A named entity is considered to be correct only if it is an exact match of the corresponding entity in the corpus.

In general, there is a trade-of between precision and recall. If the method outputs entities very conservatively, that is, reports only if it is absolutely certain of the entity, it can achieve very high precision but will probably suffer a loss in recall. On the other hand, if the method outputs entities more aggressively, then it will obtain higher recall but lose precision. A single number that captures both precision and recall is the F_1 measure, which is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

One can think of the F_1 measure as a smoothed minimum of precision and recall. If both precision and recall are high, F_1 will be high; if both are low, F_1 will be low; if precision is high but recall is low, F_1 will be only slightly higher than recall.

Chapter 4

Experiments and Results

In this chapter we compare Linear Chain Conditional Random Fields (CRF) and Maximum Entropy Model (MaxEnt) applied to the task of named entity recognition of text in the Estonian language. Firstly, we describe a set of features implemented in the system. Secondly, we study the utility of particular features by combining them into a set of experiments. Thirdly, we present the best results achieved by each method. Fourthly, we investigate the dependence of the system performance on the size of the training corpus. Fifthly, we explore the ability of the methods to correctly recognise entity bounds and analyse misclassification errors among entity classes. Next, we discuss our findings in porting the system to the sports domain. Finally, we explore impact of fine-grained language-dependent features.

4.1 Experiment Setup

In this chapter we illustrate a number of experiments in which our intention is, firstly, to compare the performance of the two systems in different conditions and, secondly, to estimate the ability of the systems to generalise to future data. To get reliable estimates, we use 10-fold cross-validation. In 10-fold cross-validation, the original text corpus is randomly partitioned into 10 subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the system, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds are then averaged for each entity class to produce the final estimation. To summarise the results of experiment with a single number, we report the weighted average of a corresponding measure over all entity classes. When splitting the data, article bounds are taken into account so that content of a single article fully falls either into validation data or training data. In this way, we minimise terminology transfer between samples used for training and testing.

A set of entity classes is limited to *locations* (LOC), *persons* (PER), *organisations* (ORG) and *facilities* (FAC). Recall, that in the text corpus, whenever two entities of the same type are immediately next to each other (e.g., *Tartust Tallinnasse*), the class of the first word of the second entity is marked with the prefix "B-". To distinguish such entities in the output of a system, we would have to introduce an additional set of at least four labels: B-LOC, B-PER, B-ORG, B-FAC. This can potentially affect system performance due to the limited amount of training data. However, we observed that only a negligible fraction (about 2%) of entities of the same type immediately follows each other in the corpus. For this reason, we treat such entities as one.

4.2 Features

The features we used can be divided into 3 classes: local, global and those derived using external knowledge. Local features are based on the token itself as well as the neighboring tokens. Global features are extracted from other occurrences of the same token in the whole document. External knowledge is represented by lists of entities collected from different sources.

4.2.1 Lists Derived from Training Data

Before we can efficiently compute some of the features, we need to process the training data and extract lists from it.

Word Prefix and Suffix List (PREF, SUF) A prefix and suffix list is compiled for each name class. These lists capture character sequences that frequently begin and terminate a particular name class. Experimentally we identified that prefixes and suffixes of length 4 and 5 produce the best results. Only those prefixes and suffixes are stored, which are associated with a unique name class and occur in the training set more than 5 times.

Word Class List (WCL) For each word in the training set we count the number of times it has been assigned a particular class. Words assigned a unique class more than 5 times are stored in this list.

4.2.2 Local Features

Lexical Feature (LEX) The string of the token converted to lower-case is used as a feature. This group contains a large number of features (one for each token string present in the training data). At most one feature in this group will be set to 1. If a token is seen infrequently during training, then it will not be selected as a feature and all features in this group are set to 0.

Word Lemma (WL) Due to abundance of inflectional endings and suffixes in the Estonian language, straightforward use of lexical features might be impractical. Firstly, treating each inflectional form of a word as a separate feature, we can easily overwhelm our system with hundreds of thousands of features. It can cause overfitting and slow convergence in training. Secondly, in order to cover words in all possible inflectional variants, we might need an unrealistically large amount of training data. For these reasons, we define this group of features which consists of lemmas, i.e. words stripped of their inflectional endings. Lemmas are obtained by passing raw text through the *t3mesta* tool. This group of features is implemented in exactly the same way as lexical features.

Capitalisation (FC) This feature is set to 1 if the word is capitalised.

First Word (FW) If the token is the first word of a sentence, then this feature is set to 1. Otherwise, it is set to 0. This feature arises from the fact that if a word is capitalised and is the first word of the sentence, we have no good information as to why it is capitalised.

Part-Of-Speech (POS) This group defines a single feature for each part-of-speech tag produced by the *t3mesta* tool. If the word w has a part-of-speech tag t, then the feature *POS-t* is set to 1.

Proper Name (PN) This features is set to 1 if the word is identified as a proper name by *t3mesta*.

Morphology (MRH) This set of features is designed to capture the constituent roots of compound words. Consider, for instance, a word *Maksuamet* composed of the roots *maksu* and *amet*. If this word is not present in the training set, it might be problematic for a system to recognise it as an organisation. However, the root *amet* can give us a right hint, as it is likely that the training data contains organisations like *Päästeamet*, *Piirivalveamet*, *Veeteedeamet*. It might be beneficial also to take into account the first root of a compound word. The prefix *lääne*, for instance, is a good indicator of a location. Two features are defined in this group which capture the first and the last root of compound words. If w has the first (last) root r, then a feature *first-root=r* (*last-root=r*) is set to 1.

Token Information (TI) This group consists of a number of features related to the character makeup of words, as listed in Table 4.1.

Group	**Feature**
Case	- Word is all uppercase - Word is mixed case
Punctuation	- Word is a punctuation mark - Internal apostrophe, hyphen or ampersand - Ends with period, has internal period - Contains semicolon
Number	- Word is a digit - Contains digit

Table 4.1: Features based on the token string.

Corpus Statistics (ST) We define two groups of features based on corpus statistics. The first group of features is designed to disambiguate words not present in the training data by analysing their prefixes and suffixes. If a token has a *n*-letter suffix (prefix), that can be found in the list SUF (PREF) for the name class *nc*, then the feature *SUF-nc-n* (*PREF-nc-n*) is set to 1. For instance, a list of frequent suffixes of a type facility includes entries *-ljak*, *-antee*, *-änav*, *-eskus*, etc. These can help to disambiguate name entities such as *Vabaduse väljak*, *Riia maantee*, *Viru keskus* and *Muhu tänav.*

The second group of features in this category checks if a word is uniquely associated with a particular name class in the training set. If a token can be found in the list WCL for the name class *nc*, then the feature *WCL-nc* is set to 1.

Surrounding Context (SC) In deciding on a class for a particular word, it is often beneficial to examine the surrounding context. Consider, for instance, the sentence fragment *Aleksandr Gerga sõnul ...*. The word *Gerga* is very unlikely to occur in the training set, nor does its string provide any clues on a correct class. However, we might observe, that in the training set the word *Aleksandr* is highly correlated with the *person* type, and that the word *sõnul* typically follows person name. Combining these facts, the correct class can be easily predicted. To expose this information, a feature set of a word is extended with binary features of its immediate neighbors. The best results were achieved using the following features: Word Lemma, Capitalisation, Proper Name and Corpus Statistics. In our example, the entity *Gerga* will be added the following features: *prev-capital*, *prev-lemma=aleksandr*, *prev-proper-name*, *prev-WCL-PER*, *next-lemma=sõna.*

4.2.3 External Knowledge

As a source of external knowledge we use a collection of lists extracted from the Web. Lists cover geographical locations, names of people, local and international organisations and facilities. The collection contains entities in both Estonian and English. Lists in Estonian were manually collected from different on-line resources. A large collection of entities in English was obtained from the web site of the Illinois Named Entity Tagger [RR09]. It contains high accuracy and high coverage gazetteers exacted from Wikipedia and other resources. Table 4.2 summarises the whole collection of lists we used.

The collection is preprocessed with the *t3mesta* tool. Words are substituted with their lemmas and turned to all lower case. Candidate words are normalised in the same way before they are matched. We employed an exact list lookup approach. For instance, the word *Pärnu* does not match list entry *Pärnu sadam*. This helps to avoid ambiguous matches. Note, that many entities are present in lists in different textual forms (e.g. Mozart, Wofgang Amadeus Mozart, Wolfgangamadeusmozart, Wolfgang Mozart, W. A. Mozart, W.A. Mozart), so exact matching should not significantly affect recall.

List Type	**Size**	**Examples**
Estonian first names [Kee]	5538	Heli-mai, Kriste, Aksenja
First names and last names in English [INE]	9348	Arnold, Yeltsin, Lee
People in English [INE]	877037	Albert Einstein, Andrus Ansip
Estonian locations [RKR]	7065	Partsi neitsijärv, Suur linnamägi
International locations [MKN]	6864	Norra, Kesk-kreeka, Pikksaar
Locations in English [INE]	5940	Los Angeles, Rio de Janeiro
Estonian organisations [EKT]	3417	Liviko, Merko Ehitus, Eesti Gaas
International organisations [INE]	329	Microsoft, Nike, Motorola
Estonian facilities [RKR]	294	Kassari sadam, Haapsalu lennuväli
Total	903573	

Table 4.2: Gazetteers, number of entries and examples.

4.2.4 Global Features

It often turns out that neither the entity string nor its context provide positive evidence of the correct entity type. In the sentence fragment *Annuk lõpetas ...* the word *Annuk* can be a person or an organisation. In this situation, it might be useful also to examine other occurrences of the word *Annuk* in a document.

For instance, observing further in the text *Andrus Annuki sõnul ...* can help us to disambiguate *Annuk* as the type *person*. To exploit this idea, we identify multiple occurrences of the same word in a document and aggregate the context the word appears in. More specifically, by context we mean here useful features of word's immediate neighbors. This includes information about word's capitalisation, its being a proper name and its membership in gazetteers. In our example, we observe that the word *Andrus* is capitalised and is present in a gazetteer of first names. We append these pieces of information to all occurrences of the word *Annuk* in a document. Context aggregation involves only those words which have been observed in a capitalised form in an unambiguous position (i.e., not in the beginning of a sentence) in a document at least once.

4.3 Feature Selection

Feature selection plays a crucial role. In our systems, we use simple count-based feature reduction. Given a threshold i, we only include those features that have been observed on the training data at least i times. Although this method does not guarantee to obtain a minimal set of features, it turned out to perform well in practice. Experiments were carried out with different thresholds. It turned out that for CRFs a threshold of 1 and 2 for MaxEnt achieved the best results.

4.4 Feature Utility Analysis

The choice of features is crucial for obtaining a good system for recognising named entities. To find an optimal set of features, in theory we would have to check all possible combinations of them. But considering the scope of the problem, this approach is computationally intractable. Instead, we have prepared a series of over 50 experiments for each system. In each experiment, a subset of features is manually selected and then the system is evaluated using 10-fold cross-validation. We start off with a very basic set of features. In every further iteration we check the utility of every candidate feature by adding it to the original set of features and re-evaluating system performance. A feature which results in the highest improvement is then preserved in the new feature set.

Experiments Table 4.3 illustrates a sequence of experiments which have led to the best results we achieved. Experiments 1 – 9 involve local features only. In the first two experiments we compared system performance relying solely on lexical features and lemmas. As we found out that lemmas significantly outperformed lexical features, the latter was excluded from the further analysis. In the following experiments we expanded the set of local features by adding information on the word's capitalisation, position in the sentence, part-of-speech, morphology, token structure, statistics and surrounding context. In the experiment 10 we inject external knowledge by adding a collection on gazetteers described in section 4.2.3. Finally, global information is added in the experiment 11.

Experiment ID	Features Used
1	LEX
2	WL
3	WL + FC + FW
4	WL + FC + FW + POS
5	WL + FC + FW + POS + PN
6	WL + FC + FW + POS + PN + MRH
7	WL + FC + FW + POS + PN + MRH + TI
8	WL + FC + FW + POS + PN + MRH + TI + ST
9	WL + FC + FW + POS + PN + MRH + TI + ST + SC
10	All local features + External Knowledge
11	All local features + External Knowledge + Global Features

Table 4.3: Description of experiments

Results Figures 4.1 and 4.2 summarise the evaluation results of the CRF and MaxEnt systems in all experiments. Results are reported as a weighted average over all entity classes. In experiments 1 and 2, where only the lexical features and lemmas are used, both systems behave quite conservatively, leaving a large proportion of entities unrecognised. As can be seen in experiment 3, information on word capitalisation and position in the sentence appears to be crucial for both systems. Both precision and recall of the MaxEnt system are largely affected by these features, while for the CRF system we mainly observe a significant gain in recall. Adding information on proper names enables the MaxEnt system to achieve comparable improvement in recall, as is shown in experiment 5. In successive experiments with the local features both systems demonstrate gradual improvement in precision and recall. Notably, adding window features in experiment 9, raises the overall performance of the MaxEnt by about 5 percent points, while affecting the CRF results less significantly. Extending the set of local features with external knowledge in experiment 10, we achieved improvement for the MaxEnt system by 2 percent points in precision and for the CRF system by 2 percent points in both precision and recall. Finally, adding global features further improves recall of CRFs by 2 percent points, while notably decreases performance of the MaxEnt system.

Summarising the results of experiments we can draw the following conclusions:

- CRFs outperform MaxEnt over all experiments.

- Competitive precision is easily achieved with just the basic set of features, while improving recall requires more elaborated information.
- Lemmas are more informative than words in their original form.
- Information on word capitalisation and position in the sentence significantly improves precision and recall of both systems.
- MaxEnt heavily relies on surrounding context and information on proper names.
- Lists of entities are equally useful for both systems.
- Global information improves recall of CRFs, but impairs the performance of MaxEnt.

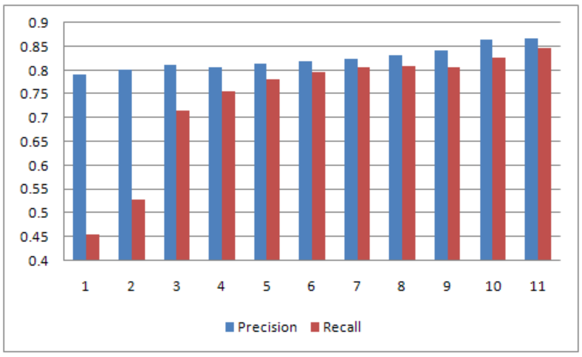

Figure 4.1: CRFs results

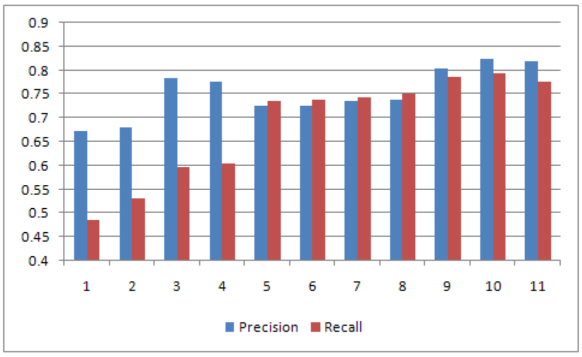

Figure 4.2: MaxEnt results

4.5 System Analysis

In the previous section we identified the best configuration for the CRF and MaxEnt models. We found out that MaxEnt reaches its peak performance using all local features and external knowledge, while CRFs in addition successfully adapt global features. In this section we analyse these two systems in greater detail.

	Precision	Recall
LOC	0.89	0.89
PER	0.89	0.89
ORG	0.82	0.76
FAC	0.77	0.56

Table 4.4: Best results for CRFs

	Precision	Recall
LOC	0.85	0.87
PER	0.86	0.81
ORG	0.77	0.71
FAC	0.61	0.38

Table 4.5: Best results for MaxEnt

Performance Tables 4.4 and 4.5 illustrate the performance of the CRF and MaxEnt systems per name class. CRFs handle *person* and *location* entities equally well in terms of both precision and recall. MaxEnt demonstrates its highest precision for the *person* type, while the *location* type has the highest recall. Recognition of *organisations* appears to be more problematic for both systems causing a notable drop in precision and recall. The lowest performance is achieved for the type *facility* (which is mostly due to the small proportion of *facility* entities in the training set, as discussed below).

CRFs significantly outperform MaxEnt over all entity classes, as is shown in Figure 4.3. Additionally, we have compared the stability of the two systems over 10 rounds of cross-validation. Figure 4.4 illustrates, that CRFs are notably more stable than MaxEnt. It implies that CRFs have advantage in distinguishing the key trends in the training data and are less susceptible to random noise.

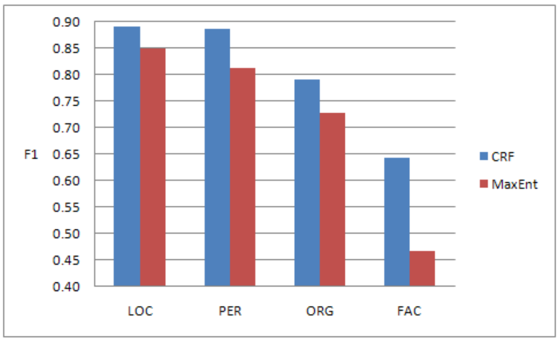

Figure 4.3: CRFs perform better than MaxEnt (t-test p-value=0.0006)

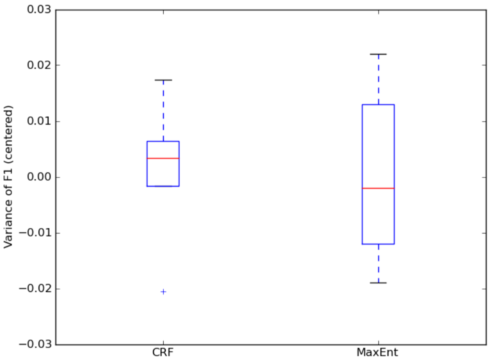

Figure 4.4: CRFs are more stable than MaxEnt

Corpus Size In this section we study the performance of the CRF and MaxEnt systems depending on the size of the training data. For this purpose, we set aside 20% of the available data for testing and organise the remaining part into 9 training datasets containing respectively 80%, 70%, . . . , 10%, 5% and 2.5% of the original data. The systems are trained on each training set and validated on the test set. Figure 4.5 illustrates the results of the experiments. Both systems demonstrate very similar learning speed: up to 1000 entities F_1 increases at a high rate, while the further increase in the size of the training material results in slower improvement. We can see that CRFs significantly outperform MaxEnt system for all entity types. The difference is especially notable for the *organisation* and *facility* types. Importantly, recognising facilities does not appear to be more complex than any other type: poor performance is due to the low proportion of *facility* entities in the training set. Following the learning trends on the plot, we can expect reasonable performance for the type *facility* starting with at least 1000 examples in the training corpus.

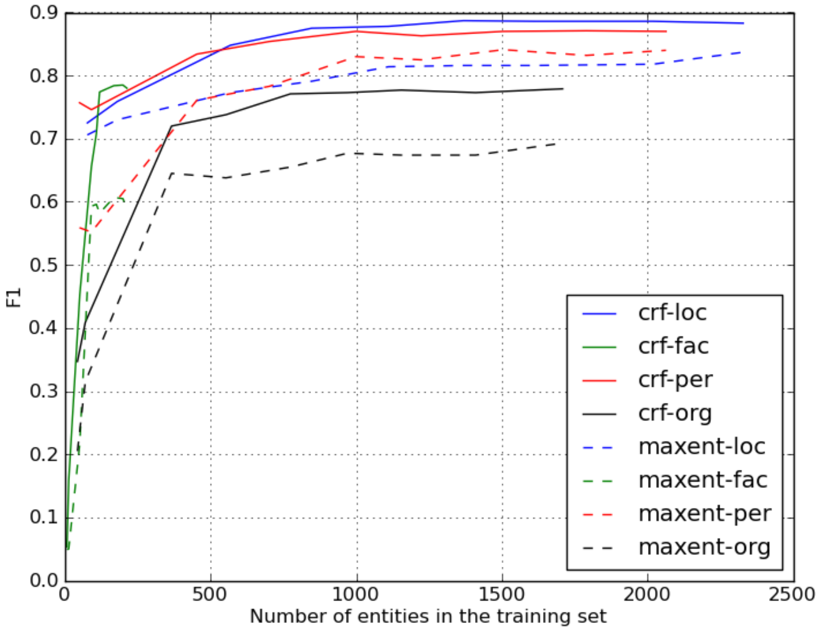

Figure 4.5: Performance of the CRF and MaxEnt systems depending on the number of entities in the training set

Exact Match Versus Word-Level Match So far, an entity has been considered correctly recognised if its boundaries exactly matched the corresponding entity in the solution (e.g., identifying a word *Korea* as a location alone will be treated as a mismatch for the entity *Korea Rahvademokraatlik Vabariik*). However, for some applications, such as text summarisation and content indexing, the constraint of exact matching is unnecessarily stringent. It is thus important to investigate the proportion of mismatches caused by boundary errors. In this experiment, we compare previously reported system performance with the results obtained using simplistic token-based matching (i.e., token is reported correctly recognised, if its predicted and actual classes match). Figure 4.6 illustrates results of a cross-validation obtained for the CRF system in its best configuration. We can see that exact matching results in a 3 percent points drop in performance on average. Recognising boundaries of *organisation* and *facility* types appears to be the most challenging.

With the token based matching, we can further investigate system performance by examining token mislabeling in the confusion matrix presented in Table 4.6. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. We observe that the system tends to confuse classes *organisation* with *location* and *organisation* with *person*. Notably, the recall of all name classes is affected mainly by falsely classifying tokens into the category *other* (i.e., not recognising them as named entities at all).

	ORG	PER	FAC	LOC	O
ORG	260	7	3	10	25
PER	12	341	3	6	13
FAC	1	1	27	2	4
LOC	12	6	4	299	7
O	38	14	8	13	8362

Table 4.6: Confusion matrix.

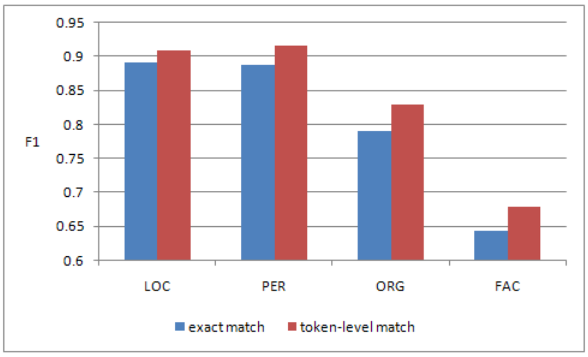

Figure 4.6: CRFs results based on exact and token-level matching.

4.6 Porting to a New Domain

As it was mentioned previously, porting the NER system to a new domain or textual genre can be challenging. The goal of this experiment is to investigate the portability of our system to the sports domain. For this purpose, we annotated a collection of 50 articles covering the most popular sport disciplines, including football, basketball, skiing, tennis and car sport. This data is used as a test set. We trained our CRF tagger in its best configuration on the whole *Delfi* dataset and evaluated on the collection of sports articles.

Table 4.7 shows the results (we excluded type *facility* as only a few instances of it were present in the collection). Note the significant drop in performance for types *person* and *organisation*. Analysing output, we found out that the system tended to heavily confuse classes *person* and *organisation*. Great proportion of misclassifications was due to names of football clubs (e.g., *Manchester United*, *Crystal Palace*, *West Bromwich*, etc.) which were falsely treated as *person* entities. Including a list of football clubs can potentially solve this problem. Low recall for entities of a type *person* is mostly owing to boundary errors. These typically occur in context where a true *person* entity is followed by a proper name not seen in the training data. Typical examples are: *Fernando Alonso Toyotal*, *Christian Klien Red Bullil*, *David Coulthard Bull-Cosworth.* This issue can be resolved by adding a number of analogous examples into the training set.

In general, the sports genre can be characterised by an extensive usage of fixed language templates (e.g., *Argentina - Jamaica 5:0*) and a limited set of domain-specific names (e.g., names of teams, competitions, sport facilities). In this perspective, the problem of NER can be effectively approached by using proper training material and custom lists of entities. To check this hypothesis, we added four representative sports articles (football, tennis, skiing, car sport) to our training set and repeated the experiment. Table 4.8 illustrates results for all entity types. Note the improvement in recall for *organisation* class from 29% to 54%.

	Precision	Recall
LOC	0.81	0.82
PER	0.70	0.57
ORG	0.43	0.29

Table 4.7: System performance on the sports domain.

	Precision	Recall
LOC	0.86	0.84
PER	0.76	0.59
ORG	0.44	0.54

Table 4.8: Results after adding four sports articles into the training set.

4.7 Language-Independent NER

The systems that we described above can be potentially used to recognise named entities in any other language as long as annotated data are provided. However, for many languages no such tools as *t3mesta* are available to perform fine-grained linguistic analysis. It means that information on the word's lemma, roots, part-of-speech and case may not be accessible. In this experiment we explore the impact of such language-dependent features. For this purpose, we set aside all the features provided by *t3mesta* and evaluate the performance of the system without those. The following set of features is used: Lexical Feature, Capitalisation, First Word, Token Information, Corpus Statistics, Surrounding Context, External Knowledge and Global Features. We use the CRF system in this experiment as it outperformed MaxEnt in all experiments and proved to be less affected by language-dependent features (see Figure 4.1). Figure 4.7 illustrates the results of 10-fold cross-validation. Results obtained by the CRF system in its best configuration are also presented in the figure for comparison. Without language-dependent features, we observe a drop in F_1 score by only 3 percent points for classes *location*, *person*, *organisation* and by 1 percent points for the class *facility.* It means, that absence of language-specific information can be effectively compensated by a combination of coarser grained local features, gazetteers and global information.

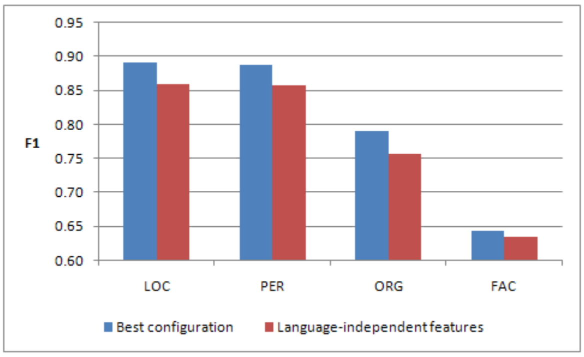

Figure 4.7: Language-independent NER

In general, however, we can not guarantee equal performance for any other language. Some features which were found useful for Estonian might be of a smaller discriminative power for other languages. For instance, we identified that information on word's capitalisation was critical for good performance (see Figure 4.1). But for the German language, where all nouns are capitalised, this feature is not informative at all.

4.8 Implementation

In the course of the work, we developed an automated Java-based environment to conveniently perform NER analysis. The environment integrates implementations of the MaxEnt and CRF algorithms and contains a number of auxiliary utilities that we used in our experiments. We used the MALLET [MAL] implementation of Linear Chain Conditional Random Fields and the the OpenNLP [ME] implementation of Maximum Entropy Models. Both algorithms are implemented in Java and are freely available on the Web. The final version of our framework is packaged with the ready-to-use CRF and MaxEnt systems which demonstrated superior performance in the course of experiments. The software is available on the CD attached to the hard copy of the thesis.

Summary

Knowledge confined within natural language can be made more accessible for machine processing by means of transforming the text into structured, normalised database form. Information Extraction aims to do just this – its goal is to automatically extract structured information from unstructured text documents using natural language processing. One basic sub-task in Information Extraction involves the recognition of predefined information units such as names of persons, organisations, locations, and numeric expressions including time, date, money and percent expressions. Named Entity Recognition (NER) is the process of identifying these entities in text. While the problem of NER has been extensively studied for widely spoken languages with the state-of-the-art systems achieving near-human performance, no research has yet been done in regards to Estonian so far.

In this work, we approached the task of recognising named entities in Estonian texts using supervised learning techniques. We explored two fundamental design decisions: choice of inference algorithm and text representation. We compared two state-of-the-art methods: Linear Chain Conditional Random Fields (CRF) and Maximum Entropy Model (MaxEnt). MaxEnt treats each word independently while allows to utilise diverse range of knowledge sources in making its tagging decisions. CRFs extend MaxEnt by exploiting the sequential nature of the problem. We studied three forms of representing named entities: 1) local features, which are based on the word itself, 2) global features extracted from other occurrences of the same word in the whole document and 3) external knowledge represented by lists of entities extracted from the Web.

To train and evaluate our NER systems, we assembled a text corpus of Estonian newspaper articles in which we manually annotated names of locations, persons, organisations and facilities.

We investigated the utility of particular features by combining them into a set of experiments. We found out that competitive precision can be easily achieved with just the basic set of features, while improving recall requires more elaborated information. Lemmas appeared to be more informative than words in their original form. Information on word capitalisation and position in the sentence significantly improved precision and recall of both systems. Surrounding context and information on proper names proved to be especially important for MaxEnt.

Lists of entities were equally useful for both systems. It turned out that global information improved recall of CRFs but impaired performance of MaxEnt.

The experiments demonstrated that CRFs significantly outperformed MaxEnt over all entity classes. We report the best result of 0.86 F_1 score achieved by CRFs using combination of local and global features and external knowledge.

We explored the portability of our system to the sports domain. As a result, we obtained significant drop in performance for types *person* and *organisation*. Most misclassification errors were due to domain-specific entities not observed in the training data and gazetteers. By adding just a handful of sports articles into the training set we achieved a significant improvement for a class *organisation*.

Finally, we showed that elimination of morphological and grammatical information did not significantly affect system performance. It means, that our system can be used to recognise named entities with any other language even if no tools are available to perform fine-grained linguistic analysis.

Nimega üksuste tuvastamine eestikeelsetes tekstides

Magistritöö (20ap)

Aleksandr Tkatšenko

Resümee

Käesoleva töö raames uuriti eestikeelsetes tekstides nimega üksuste tuvastamise probleemi (NÜT) kasutades masinõppemeetodeid. NÜT süsteemi väljatöötamisel käsitleti kahte põhiaspekti: nimede tuvastamise algoritmi valikut ja nimede esitusviisi. Selleks võrreldi maksimaalse entroopia (MaxEnt) ja lineaarse ahela tinglike juhuslike väljade (CRF) masinõppemeetodeid. Uuriti, kuidas mõjutavad masinõppe tulemusi kolme liiki tunnused: 1) lokaalsed tunnused (sõnast saadud informatsioon), 2) globaalsed tunnused (sõna kõikide esinemiskontekstide tunnused) ja 3) väline teadmus (veebist saadud nimede nimekirjad).

Masinõppe algoritmide treenimiseks ja võrdlemiseks annoteeriti käsitsi ajakirjanduse artiklitest koosnev tekstikorpus, milles märgendati asukohtade, inimeste, organisatsioonide ja ehitise-laadsete objektide nimed.

Eksperimentide tulemusena ilmnes, et CRF ületab oluliselt MaxEnt meetodit kõikide vaadeldud nimeliikide tuvastamisel. Parim tulemus, 0.86 F_1 skoor, saavutati annoteeritud korpusel CRF meetodiga, kasutades kombinatsiooni kõigist kolmest nime esitusvariandist.

Vaadeldi ka süsteemi kohanemisvõimet teiste tekstižanridega spordi domeeni näitel ja uuriti võimalusi süsteemi kasutamiseks teistes keeltes nimede tuvastamisel.

References

[ACE] Automatic Content Extraction (ACE) Evaluation – offcial website. `http://www.itl.nist.gov/iad/mig//tests/ace/` (accessed on May 24, 2010).

[AM03] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 8–15, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[BBN] ANNOTATION GUIDELINES FOR ANSWER TYPES – offcial website. `http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html` (accessed on May 24, 2010).

[BdM^{+}92] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, 1992.

[BMSW97] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[BONV03] Oliver Bender, Franz Josef Och, Hermann Ney, and Lehrstuhl Für Informatik Vi. Maximum entropy models for named entity recognition. In *Proceedings of CoNLL-2003, 148Ű151*, pages 148–151, 2003.

[Bri98] Sergey Brin. Extracting patterns and relations from the world wide web. In *In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBTŠ98*, pages 172–183, 1998.

[BSAG98] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *In Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998.

[CN02] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: A maximum entropy approach using global information. In *In Proceedings of COLING02*, pages 190–196, 2002.

[Col02a] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. pages 1–8, 2002.

[Col02b] Michael Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. pages 489–496, 2002.

[CS92] Sam Coates-Stephens. The analysis and acquisition of proper names for the understanding of free text. *Language Resources and Evaluation*, 26(5):441–456, December 1992.

[CS04] William W. Cohen and Sunita Sarawagi. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, New York, NY, USA, 2004. ACM.

[DFM+04] Shipra Dingare, Jenny Finkel, Christopher Manning, Malvina Nissim, and Beatrice Alex. Exploring the boundaries: Gene and protein identification in biomedical text. In *In Proceedings of the BioCreative Workshop*, 2004.

[ECD+05] Oren Etzioni, Michael Cafarella, Doug Downey, Ana maria Popescu, Tal Shaked, Stephen Soderl, Daniel S. Weld, and Er Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134, 2005.

[EKT] Eesti Kaubandus-Tööstuskoda.
`http://www.koda.ee/?id=1916` (accessed on May 24, 2010).

[FH02] Michael Fleischman and Eduard Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[FIJZ03] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *In Proceedings of CoNLL-2003*, pages 168–171, 2003.

[Fle01] Michael Fleischman. Automated subcategorization of named entities. In *ACL (Companion Volume)*, pages 25–30, 2001.

[GAT] GATE – offcial website.
`http://gate.ac.uk/` (accessed on May 24, 2010).

[GS96] Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[HJK98] Tarmo Vaino. Heiki-Jaan Kaalep. Kas vale meetodiga õiged tulemused? statistikale tuginev eesti keele morfoloogiline ühestamine. *Keel ja Kirjandus*, pages 30–38, 1998.

[HvdB03] Iris Hendrickx and Antal van den Bosch. Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data. In *In Proceedings of CoNLL-2003*, 2003.

[INE] Illinois Named Entity Tagger – offcial website.
`http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJNE` (accessed on May 24, 2010).

[Jay57] Edwin T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[Kee] KeeleWeb.
`http://no.spam.ee/~kaur/keelewebi_eesnimed`
(accessed on May 24, 2010).

[KGW$^+$95] Morgan Kaufmann, R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie system as used for muc-6, 1995.

[KT07] Jun'ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[KTK07] Roman Klinger, Katrin Tomanek, and Roman Klinger. Classical probabilistic models and conditional random fields, 2007.

[LGL05] Seungwoo Lee and G. Geunbae Lee. Heuristic methods for reducing errors of geographic named entities learned by bootstrapping. In *Proc. International Joint Conference on Natural Language Processing.*, 2005.

[Lia05] P Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2005.

[MAL] MALLET: A Machine Learning for Language Toolkit – offcial website.
`http://mallet.cs.umass.edu/` (accessed on May 24, 2010).

[Mcd96] David D. Mcdonald. Internal and external evidence in the identification and semantic categorization of proper names. In *Corpus Processing for Lexical Acquisition*, pages 21–39. MIT Press, 1996.

[MD03] Fien De Meulder and Walter Daelemans. Memory-based named entity recognition using unannotated data. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003*, pages 208–211, 2003.

[ME] OpenNLP MaxEnt Package – offcial website.
`http://maxent.sourceforge.net/` (accessed on May 24, 2010).

[MGZ04] Scott Miller, Jethran Guinness, and Alex Zamanian. Name tagging with word clusters and discriminative training. In *Proceedings of HLT*, pages 337–342, 2004.

[Mik99] Andrei Mikheev. A knowledge-free method for capitalized word disambiguation. In *In 37th Annual Meeting of the Association for Compuational Linguistics*, pages 159–166, 1999.

[MKN] Maailma kohanimed.
`http://www.eki.ee/knab/mkn_ind.htm` (accessed on May 24, 2010).

[ML03a] Andrew Mccallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

[ML03b] Andrew Mccallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

[MR04] Sven Mika and Burkhard Rost. Protein names precisely peeled off free text. *Bioinformatics*, 20(1):241–247, 2004.

[MSC] Morpho-syntactic categories.
`http://www.cl.ut.ee/korpused/morfliides/seletus.php?lang=en` (accessed on May 24, 2010).

[MTU+01] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named entity recognition from diverse text types. In *In Recent Advances in Natural Language Processing 2001 Conference, Tzigov Chark*, 2001.

[MWC05] Einat Minkov, Richard C. Wang, and William W. Cohen. Extracting personal names from email: applying named entity recognition to informal text. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 443–450, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[NRVsAs03] M. Narayanaswamy, K. E. Ravikumar, K. Vijay-shanker, and K. Vij Ay-shanker. A biological named entity recognizer. pages 427–438, 2003.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[RJ99] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999.

[RKR] Riiklik Kohanimeregister.
`http://www.eki.ee/knab/mkn_ind.htm` (accessed on May 24, 2010).

[RR09] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[RTWH00] Thomas C. Rindflesch, Lorraine Tanabe, John N. Weinstein, and Lawrence Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. volume 5, pages 514–525, 2000.

[Sek98] Satoshi Sekine. Nyu: Description of the japanese ne system used for met-2. In *Proc. Message Understanding Conference*, 1998.

[Set04] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA*, pages 104–107, 2004.

[SN] Satoshi Sekine and Chikashi Nobata. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy.

[SZZ$^+$03] Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew lim Tan. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *In: Proceedings of NLP in Biomedicine, ACL*, pages 49–56, 2003.

[Thi95] Christine Thielen. An approach to proper name tagging for german. In *In Proceedings of the EACL-95 SIGDAT Workshop: From Text to Tags*, 1995.

[TKSDM03] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[TM] A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *In EACL 2006.*

[TT03] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *Proc. of the ACL-03 Workshop on Natural Language Processing in Biomedicine*, pages 41–48, 2003.

[VS07] Dániel Varga and Eszter Simon. Hungarian named entity recognition with a maximum entropy approach. *Acta Cybern.*, 18(2):293–301, 2007.

[Zho04] Shen D. Zhang J. et al. Zhou, G. Recognition of protein/gene names from text using an ensemble of classifiers and effective abbreviation resolutionzhou, g., shen, d., zhang, j. et al. In *Proceedings of the EMBO workshop BioCreative: Critical Assessment for Information Extraction in Biology, 28thŰ31st March, Granada, Spain*, 2004.

[ZWB+99] Ian Witten Zane, Ian H. Witten, Zane Bray, Malika Mahoui, and W. J. Teahan. Using language models for generic entity extraction. In *In International Conference on Machine Learning Workshop on Text Mining*, 1999.

Appendices

Appendix A. Program code (on a compact disc)