# Cross-Document Named Entity Co-reference Resolution for Dutch

*As a pre-process for named entity based text mining*


Corné Versloot

October 24, 2007

# Cross-Document Named Entity Co-reference Resolution for Dutch

*As a pre-process for named entity based text mining*

Master Thesis of Corné Versloot

March 2007 - October 2007


University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science.

Department of Human Media Interaction

P.O. Box 217
7500 AE Enschede

Supervisors:
ir. R.B. Trieschnigg
prof. dr. F.M.G. de Jong

TNO Information and Communication Technology.

Department of Broadband and Voice Solutions

Brassersplein 2
P.O. Box 5050
2600 GB Delft

Supervisor:
dr. ir. W. Kraaij

# Preface

This thesis is the result of my graduation project performed at TNO-ICT in the department of Broadband and Voice Solutions. In this graduation project I performed research into named entity co-reference resolution for Dutch and the influence of this process on name based text mining. With this thesis I conclude the computer science master 'Human Media Interaction' which I followed at the University of Twente.

During this work I was supported by Wessel Kraaij, Dolf Trieschnigg and Franciska de Jong. I would like to thank them for all their support. I would also like to thank the people in the department and especially my direct colleagues in room BA301 for their support and interest in my work.

# Abstract

People create massive amounts of texts, finding information within these texts usually involves reading. It is possible to extract information from texts automatically but most techniques are farm from perfect. One example is text mining based on names: extraction of names from texts and discovery of information using these names. However, names often have a lot of variances, the same name can refer to different things and different names can refer to the same thing. Finding out which names in a large set of documents refer to which 'entities' in the world is the focus of this graduation project.

This research studied methods to perform cross-document named entity co-reference resolution for Dutch and the impact of this resolution on name based text mining. In order to do this the named entity co-reference resolution was split into two separate (sequential parts) parts: co-reference resolution in single documents and co-reference resolution in multiple documents.
The single document study was heavily based on surface form similarity of names following the hypothesis that similar names within one document are co-referential. Two methods were created and evaluated with respect to a baseline method.
The co-reference 'clusters' created by the best of these two methods served as a starting point for cross-document named entity co-reference resolution. In this part of the research machine learning techniques were used to train a model capable of distinguishing co-referential names from non-co-referential names.
The best method from this study was used to pre-process news articles for Novalink, a text mining tool developed by TNO based on named entity co-occurrence. Two versions of Novalink, one with and one without named entity co-reference resolution were compared in an evaluation to study the impact of named entity co-reference resolution.

A few important conclusions can be derived from the evaluation:

- Single document NE co-reference resolution can be done using only name similarity.
- Similarity of names is very important to solve NE co-reference resolution, but more information is needed to solve 'hard' cases (especially in cross-document NE co-reference resolution).
- High recall scores of the groups of co-referential names does not directly lead to 'better' text mining results (In case of the Novalink tool). Good results can be obtained using less than the optimum number of names.
- It is important that groups of co-referential names have a high precision to exclude erroneous results.

# Contents

# 1 Introduction

People generate lots of information every day. Information can have various forms like news broadcasts, news articles, web sites, radio, web logs etc. A lot of this information is freely available on internet or is contained in large databases or storage facilities (news paper archives, TV archives etc.). Access tools for these archives allow people to search for sources using key words. Finding the specific answers or the information you are looking for typically involves reading (or watching/listening).
An alternative is to extract and summarize information from texts automatically to provide the user with a global overview and starting point for further search. This functionality is studied in the field of text mining, a research field that develops and evaluates methods to automatically find information in written documents (a more formal definition is given in chapter 2). Useful information can be found in patterns in texts, or extracted from multiple documents. For example names that frequently co-occur in texts is a pattern. In the case of names it can be said that the entities the names refer to have some kind of relation.
This is basically what Novalink does. The purpose of the tool is to give an overview of named entities (NE) that are related to another NE. Named entities are things (objects, persons, organizations etc) in the real word which are referred to by names or noun phrases in texts. Novalink tries to find relations between named entities in news articles based on their co-occurrence. More precisely; NE's that frequently occur in the same document, more than can be expected by chance, are assumed to have 'something to do with each other'. An example of a Novalink result is shown in figure 1.1. A more thorough description of this system can be found in chapter 3.
Novalink is just one example of a text mining tool. The field of text mining is relatively young and a lot of work is needed in order to reach the fields 'full' potential. My graduation project described in this thesis is just one small step in this direction.

## *1.1 Research questions*

### 1.1.1 Problem description

From a computational point of view texts are just simply sequences of characters that do not have any meaning. These sequences can be broken down into pieces such as words, sentences and paragraphs. Humans have a lot of knowledge concerning the meaning of these pieces and that enables them to understand the meaning of texts. Computers lack this knowledge making it very hard to understand even the simplest sentences, not to mention entire texts.

One important piece of information for name based text mining systems is which names refer to the same entity. Names that refer to the same entity are co-referential. For example 'George Bush' and 'G.W. Bush' are co-referential because they both refer to the American president (in certain temporal context). Unfortunately systems often lack this kind of information. Novalink uses a very simple rule to determine if names are co-referential: *names that are exactly the same co-refer to the same entity*.
The result of this rule is then used for text mining, resulting in imperfect results. Some of the mistakes made by the system as a result of this rule can be seen directly in the result; other mistakes can not be seen but can be

found by simple reasoning. The different types of mistakes are described in the following list:

- Synonymic names (different names for the same entity) are treated as separate entities which can cause the system to miss relations. An example can be the entity 'Henk Kamp' with two name variations 'Kamp' and 'Henk Kamp'. When searching for 'Henk Kamp' the possible entities that often co-occur with 'H. Kamp' can be missed.
- Homonymic names (that are exactly the same) that refer to different NE's are treated as one entity. The result is that relations found for this entity can in reality belong to different entities and thus give an erroneous result. An example that really occurs within Novalink has to do with the name 'Marco', the first name of a soccer coach (van Basten) and the first name of a singer (Borsato). When searching for Marco the relations that are found are from both people giving a very strange result with relations from both the music and from the soccer domain.

Users generally can not detect these mistakes because they do not know anything about the NE of interest and hence can not evaluate the correctness of the result. The only effect of the problem thing that does show up in the results is 'double entities'. These are entities that are co-referential in reality but are separate entities within the result. Novalink often thinks that these entities have a relation because they frequently co-occur in documents. An example of this can be seen in figure 1.1 where there are a total of 5 nodes that refer to Beatrix. This graph also shows that not only names refer to NE's but also noun phrases like 'Hare Majesteit de Koningin'.

The problems described above would not occur if the system knows exactly what names belong to what entities. Finding this information in multiple texts is called 'cross-document named entity co-reference resolution'. It is needed to solve the problem for names from multiple texts because Novalink works on a large set of documents. Simply said the task is to find out for all the names in a set of documents what entity they refer to; if multiple names refer to the same entity they are co-referential.

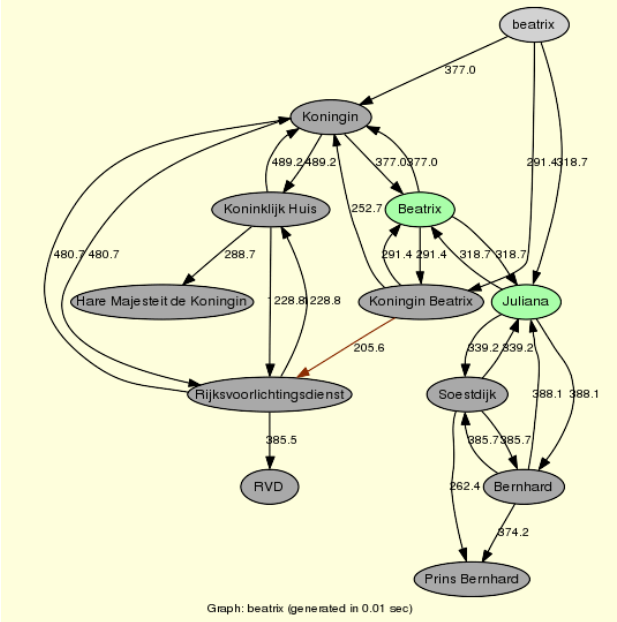



**Figure 1.1**: A Novalink result for 'beatrix' with 5 different nodes that refer to the Dutch queen

Finding all the co-referential names can be a very time consuming process and the document collection for which this technique can have added value may be huge. Because of this the method(s) used to solve the problem must be fast and scalable to process all the documents in an acceptable amount of time. Scalable means that when the number of documents doubles the processing time should not grow unacceptably i.e. 'explode'.

### 1.1.2 Main research question

First the main research question and hypothesis for this research are formulated. Then this question is broken down into a number of sub-questions used to focus on different aspects of this research.

The main research question is:
*Can Dutch text mining be improved using named entity co-reference resolution?*

The hypothesis concerning this question is:
*A text mining system that has information about NE co-references will give better results than a system without this knowledge.*

I expect a system that uses NE co-reference resolution to perform better than a text mining system without this knowledge. The reason is very simple: understanding which names refer to which NE's enables to system to find better results (relations in case of Novalink). With 'better' is meant less double entities and more correct relations (see also the requirements for the Novalink evaluation in chapter 8.1).

### 1.1.3 Sub questions

To answer this research question a set of sub questions needs to be solved or answered. An important part of the research is to develop an adequate NE co-reference resolution method.

*What information (features) can be used to solve cross-document NE co-references.*
The most basic solution to the problem is to use string matching techniques to asses whether two names are co-referential or not. However, string matching techniques are not enough to solve all NE co-references, especially when names are homonymic/ambiguous or synonymic more information is needed. Features could be extracted from texts (important nouns, important names etc) themselves and from metadata about the texts (author, publish date etc). The question is what information could be used and how informative (with respect to the problem) they are. A lot of information on this matter can be found in literature.

*What sub-set of these features gives the best co-reference resolution result with a minimum of computation complexity?*
As described above, there could be a lot of different features that can be used to solve NE co-references. However, it is likely that some features are more important than others. Secondly, the extraction of some features might be more expensive than the extraction of others. Since the final solution needs to be scalable there is a trade-off between the features to use (and the quality that can be accomplished using these features) and the total computational complexity.

*How exactly does the NE co-reference resolution effect the text mining task?*
This question is not the same as the main research question but addresses a more specific point; *how* is the text mining affected. Introducing a NE co-reference resolution method in a text mining system will have consequences and the question is what these are exactly. It might for example be possible that changes need to be made to the text mining method to make better use of the NE co-reference resolution.

### 1.1.4 Restrictions

This paragraph describes pragmatic restrictions used to limit the scope of the research, explains where the restrictions come from and how they influence this work.

*Restriction 1: The Novalink Named Entity Recognizer (NNER)*
Novalink uses a name recognizer to recognize the names in the articles. It is not part of this research to alter or improve this recognizer. As a result this study is based on the names recognized by the. What names the NNER recognizes is described in chapter 3.2.
Since methods created in this study are tuned to the NNER output it is possible that they show different performance when working with names recognized by other programs.

*Restriction 2: No usage of syntactic and semantic knowledge*
Syntactic, semantic and world knowledge can be very useful in this research and are frequently used in for example 'anaphora resolution' (see chapter 2). Unfortunately these techniques are often computational expensive and correct usage of them for Dutch is a study on its own. As a result these techniques are not used in this research. The restriction to use a 'knowledge poor' approach is possibly a lower performance on the created methods (in comparison with knowledge rich methods). More information regarding syntactic, semantic and world knowledge can be found in chapter 2.6.

## *1.2 Methodology*

First a literature study was done to find general information regarding the research questions and relevant research fields (NLP and text mining). The findings of this study formed the basis and general guideline for this graduation project. A corpus containing cross-document NE co-reference annotations was created for Dutch and was used as ground truth data for evaluations.
The NE co-reference resolution problem was split into two parts: single document and cross document NE co-reference resolution (inspired by David Yarowsky (Yarowsky; 1995)). A baseline method was created for single document NE co-reference resolution. More advanced methods were build in an iterative way and evaluated with respect to the baseline.
The NE co-references found within single documents served as the basis for the development of cross-document named entity co-reference resolution methods. The methodology for the cross document co-reference resolution was largely the same as for the single document resolution. A baseline method was developed and more advanced methods were built and evaluated. The best method was implemented into Novalink and the results were compared with results from the 'unaltered' Novalink. This evaluation indicates the usefulness of the NE co-reference resolution done for text-mining and thus answers the main research question.

## *1.3 Thesis overview*

The table of contents and the methodology section already gave an indication of the steps used to answer the research questions. This section will describe the different chapters in this document in more detail.
The second chapter provides the state of the art knowledge that served as a basis for this research. The chapter is split up into five different sections: definition of text mining, explanation of MUC, work done using named entities, previous work on co-reference resolution and some background about anaphora resolution. Most important aspects of these two fields and their usage for this research will be discussed there.

The purpose and internal mechanics of Novalink are described in chapter 3. This chapter also contains the methodology used to evaluate the NNER and an assessment how the NNER influences this research.
A good evaluation method was needed to assess the quality of the developed methods. It turned out that there was no such method available, at least none that fulfilled the requirements for this research. Chapter 4 describes different evaluation methods with their strong and weak points and describes the final choice used in this research.
The evaluation method needed ground truth data, in this case a corpus with cross-document named entity annotations. Chapter 5 describes the work done to create this corpus. This chapter also contains examples of hard NE co-reference cases to give an indication of the difficulty of the problem.
As described in the methodology section it is common to start with single document named entity co-reference resolution. The definition of a baseline method and the iterative process to create and evaluate new methods is described in chapter 6. The results from the single document method serve as a baseline for the cross-document named entity co-reference resolution methods described in chapter 7. This chapter describes the different features that are extracted and the machine learning process used to build a good model. Most answers to the sub research questions will be given in this chapter.
The last step of the research is the evaluation of the effect of NE co-reference resolution on name based text mining. This evaluation is done using the 'old' and 'new' Novalink systems. The exact evaluation methodology and the results are described in chapter 8.
The last three chapters; 9, 10 and 11 respectively contain the discussion, conclusion and future work. These chapters will give the answer to the research questions and will look back at the research to point out aspects that can be improved. The future work section describes possible next steps in the research of NE co-reference resolution.

# 2 Related Work and State of the Art

## *2.1 Introduction*

This chapter provides the setting of this research; what work has already been done and what are useful methods that can be used in this research. The chapter is divided into five sections:

- What is text mining? This section describes the differences between data mining, text mining and information retrieval.
- Message Understanding conference: the most influential conference in the field of Information Extraction (IE) the field containing data- and text-mining.
- Named entities: definitions and techniques that are useful for this research.
- NE co-reference resolution: a section explaining common methods to solve NE co-references.
- Anaphora resolution: part of Natural Language Processing that studies co-references in broader form.

## *2.2 What is text mining?*

Data mining and text mining (related to information retrieval) are mentioned in the introduction and here I clarify what they exactly mean and how they relate to this research. This section will explain the terminology and will use some examples for clarification. The ideas and definitions described in this section are mainly derived from work done by Hearst (Hearst; 1999).

Hearst et al. introduce a number of parameters used to distinguish between the different types of 'mining':

- Novelty of information:
  - finding patterns
  - finding novel information
  - finding non-novel information
- Type of data:
  - Structured data or strictly formatted text (data in databases).
  - Textual data: normal (unstructured) text.

An important parameter is the novelty of information; did a system really find 'new' information or did it not (for example finding information sources). With new information Hearst means latent information; knowledge that is present in some way but is not used explicitly. Using these parameters Hearst proposes the categorization as described in table 2.1.

| | **Finding patterns** | **Finding novel information** | **Finding non-novel information** |
|---|---|---|---|
| **Structured data** | Data mining | ? | Database queries |
| **Textual data** | Computational linguistics | Text mining | Information retrieval |

**Table 2.1**: overview of data and text mining tasks as proposed by Hearst et al.

This table shows that data mining is 'defined' as the search for patterns in structured data. Hearst gives the example of mining a supermarket database for products that are often sold together (so they can be put together in the same alley). Word co-occurrence is used as an example of the discovery of patterns in textual data as covered by the field of computational linguistics. Text mining is defined as finding novel information in textual data.

Following these criteria Novalink would not be a text mining tool but would do 'computational linguistics' because it finds a pattern (frequently co-occurring names). In my opinion this has to do with the definition of patterns; what are patterns exactly? Hearst only mentions patterns in words and claims that "computational linguistics applications tell us about how to improve language analysis, but they do not discover more widely usable information". In my opinion this depends on the patterns that are discovered and on possible interpretation of patterns. The patterns found by Novalink do not improve language analysis but provide the user with new knowledge that was not written down intentionally by the author(s). From that perspective Novalink is a text mining tool.

## 2.3 Message Understanding Conference

The Message Understanding Conference, more commonly referred to as MUC, is the most important Information Extraction. MUC is mostly an evaluation framework to compare and benchmark different methods built for different tasks. The conferences are numbered, starting with MUC-1 in 1987. The first two conferences were initiated by Beth Sundheim and focused on knowledge extraction from military messages. Throughout the years MUC supported a growing number of tasks such as: named entity recognition, attribute extraction, relation extraction and a few more. A cross document named entity co-reference resolution task was proposed for MUC-6 and an evaluation method was defined (see section 4.3.1). However it was not included as a formal task because it was considered to be too ambitious at the time (Grishman; 1994).
MUC provides datasets for the tasks which are evaluated at the conference. A lot of the IE extraction methods known today were firstly created to solve MUC tasks. The last few years also work on other languages or combination of languages has been added to MUC under the name of Multilingual Entity Task (MET).
Data sets for MUC-6 and MUC-7 were created by the Linguistic Data Consortium (LDC). The LDC named entity annotation guide served as a bases for the creation of the corpus in this research (section 5.3). Some of the methods and evaluation metrics made for MUC tasks were used in this graduation project.

Closely related to MUC is the 'Conference on Natural Language Learning' (CoNNL) an annual event first organized in 1997. The CoNLL conference has one shared task every year, the participating systems for this task are evaluated and compared in a systematic way (like in MUC). On of the tasks was multi language named entity recognition. The dataset for this task was used as training set for the NNER.

## 2.4 Named Entity Analysis

Novalink's text mining is entirely built upon named. But what are names and named entities exactly? Many different aspects of names have been studied: what defines named entities, how do they evolve throughout a text, how can they be recognized and more. This section describes the most important work done with named entities.

The terms named entity, name and other associated terms are used a lot in this report and hence it is important to have a clear understanding these terms:

- Named Entity: an entity in the real world that can be referred to by using proper names or definite descriptions. Named entities are mostly physical objects but they can also be more abstract concepts (like theories).
- Proper noun/name: a noun that denotes a particular NE; usually capitalized in European languages. Simply 'name' will be used throughout this report to denote these proper nouns/names.
- Definite description: a noun-phrase that refers to a specific NE (like mayor of Amsterdam).

### 2.4.1 Rigid- and non-rigid designators

Two classes of names and noun phrases were first defined by the philosopher Saul Kripke in 'Naming and necessity', three lectures on reference of proper names. These classes are rigid- and non-rigid designators and are important for this research. A rigid designator is a proper name that depending on the context refers to the same entity. A non-rigid designator is usually a definite description that does not refer to the same entity even if the context is the same. For example '*the mayor of Amsterdam'* is non-rigid because it does not refer to one unique person (the mayor in 1980 was someone else as the mayor in 2000).
Rigid and non rigid designators can be found throughout the news articles. The mayor example above surely refers to a real person and using extra information like the publication date the referent can be identified. Without usage of semantic information (restriction number 2) non-rigid designators are very hard to solve. Some people are often referred to by non-rigid designators, often using their job titles (Dutch soccer coach, prime minister). These NE's often contain useful information from text mining point of view and hence are important to process.

### 2.4.2 Name recognition

One of the first MUC tasks was the recognition of names, a challenging task with a lot of exceptions. For English and Dutch names are capitalized which is a useful feature for name recognition. However, not all capitalized words not at the beginning of a sentence are names, and words at the beginning of a sentence can be names. Also, names can contain words that are not capitalized, can be concatenated together or can contain abbreviations. It is hard to recognize or exclude all the different exceptions.
In general there are two ways to do name recognition: rule based (using various kinds of information) and machine learning. Rule based recognizers use hand crafted rules which are often typical for the language. Rules give an indication if a word (group) is likely to be a name. Commonly used rules for English and Dutch are:

- Word must be capitalized
- Capitalized word followed by another capitalized word is often one name
- Capitalized words in subject role are often names

Rule based systems typically use the surface form of the words and syntactic analyzers (like part-of-speech taggers). Machine learning techniques use annotated data to learn the characteristics of names in a certain language. These characteristics can very from surface forms to syntactic or even semantic information.

Examples from (Louis A., de Waal A, C. Venter; 2006) are:

- Word capitalization and word length
- PoS tags
- Possessive ending (Jane**'s**)
- Company information (Ltd, TM etc)
- Titles (Mr, lord, professor etc)

Sometimes rules are used in a post process to correct some of the (mostly simple) mistakes made by a learned model. Both types of name recognition methods often use pre compiled lists of names called gazetteers. It is also possible to use multiple different methods in a sequence to 'minimize' the effect of weak points of individual recognizers. Radu Florian explored this approach for the Conference on Natural Language Learning (CoNLL) NE recognition task and reports an accuracy around 98% (Florian R.; 2002).
CoNLL also has a language independent NE recognition task which is changing since language specific information can not be used. Sang et al. describe the sixteen systems and their performance of this task (for English and German). The best performing method by Florian et al. (Florian, Ittycheriah, Jing, and Zhang; 2003) achieved an F-measure of 88.8% for English and 72.4% for German.
The CoNLL data sets are pre-processed with an IOB tagger. This tagger tags words with either an I (inside), O (outside) or B (begin). Names can be extracted using sequences of these tags: a B followed by multiple I's. In some variations of the IOB tag set the B is omitted, the first occurrence of an I after an O can then be seen as the old 'B' tag. The IOB tag set was first defined by Ramshaw et al. (Ramshaw and Marcus; 1995).

### 2.4.3 Classification of names

Another thoroughly investigated aspect of names is their types. The type of a name is what the entity is that is referred to by the name. Most common types used within research are person, organization and location. However this is not a complete set. The types as used for MUC, described by the Linguistic Data consortium are: person, organization, facility, location, Geo-political entity, vehicle and weapon (Linguistic Data Consortium; 2005). One example of work done in this direction is the identification of types in Korean news articles by Kim et al. (Kim, Kang and Choi; 2002). They report 73.16% precision and 72.98% recall for 2580 types. Type information is very useful for information extraction since it adds semantic knowledge to names. The types normally used are not always sufficient and more types or hierarchical classification gives a lot more useful information. One example is work done by (Fleischman and Hovy; 2002) who develop a method to distinguishing three person types (politician, entertainer, businessmen) using the local context of the names.
Knowing the type of a named entity can be very useful for the disambiguation of similar names. This is the reason that types of names were also annotated in the corpus and their usefulness was studied (section 6.5).

### 2.4.4 Evolution of names in news articles

A very interesting study was done by Nenkova and McKeown (Nenkova and McKeown; 2003). They studied occurrences and modifications of names throughout English news articles. Their main focus was on the introduction of names in a text and if these names are changed in subsequent occurrences (and if so, in what way). They found that:

- names are initially used fully (with or without pre-modifiers like tiles)
- this initially used name is very likely to be changed since the formal name initially introduced is not suited for continuous usage (happens 76% of the cases)
- if a modified form is chosen it is not likely the name will change more
- Usage of first names and nick names is very unlikely, but if they are used they will probably be continuously used throughout the remainder of the text.
- Normally subsequent mentions of names have less or equal modifiers (such as Mr. and Mrs.). In case of one modifier the chance is 50% it will be used again.

This information is interesting because it describes how names can evolve in texts and the chance that certain changes occur.

## *2.5 Named entity co-reference resolution*

Named entity co-reference resolution is the process of finding co-referential names in groups of texts. The techniques used to do this largely depend on the domain of the texts and availability of certain knowledge (like syntax and world knowledge). Information that is frequently used is: lexical knowledge, domain knowledge, context and world knowledge. The following sections explain these concepts in more detail. However, some techniques are quite complex and the reader is advised to read the referred work to get a better understanding of the different techniques.

### 2.5.1 Lexical knowledge

Lexical knowledge is always needed when performing NE co-reference resolution. Different names need to be compared in a meaningful way to give an indication if two names are co-referential. There are a lot of different string matching (similarity assessment) techniques that can be used. Cohen (William) did a lot of research in this direction. The research done by Cohen et al. (Cohen, Ravikumar and Fiendberg; 2003) compares string matching techniques for name matching purposes. Edit-distance, token based distance and hybrid distance functions were compared. Their conclusion is that a combination of JaroWinkler token matching with TFIDF ranking scheme performs best (slightly better than these methods used independently).

### 2.5.2 Using meta-data

Meta data is specific knowledge about the texts that can be used for the NE co-reference resolution process. Meta-data is often supplied along with texts or can be extracted relatively simply (recognition of titles, authors etc.). The data available might differ depending on the type of text. Examples are:

- Author(s)
- Title
- Publish date
- Category of text (sports article, political article, action, humour etc)
- Publisher
- Keywords (for example in scientific articles)

There can be a lot more information like this depending on the type of text and the information associated with them.

An example of work that heavily depends on this type of knowledge is research done by Lee et al. (Lee, On, Kang and Park; 2005). This research tries to fix citation problem in digital libraries. These libraries have a large number of citation records with a lot of names which can contain spelling errors or are similar to other names. Lee et al. describe two different problems: split citation (one author with different name spelling variants) and mixed citation (multiple authors that have the same name). This is the same problem as described in this research (although in a different domain). To compare two citations 'article meta-data' like co-authors, titles and venues are extracted from the database and represented as vectors. The similarity between these vectors is calculated to find out if the two citations belong to the same author or not. They report an overall accuracy of 90%-93% (precision and recall results are not presented).
It should be noted that this system is not a text mining system since it does not extract the information from raw texts but from a database. Because of this the information the system can work with is very accurate.

### 2.5.3 Using context

Lexical information from names is always needed but does not always solve NE co-references correctly, very common names are often hard to solve correctly. The context of the names can be used to help solve these hard cases (and solve 'easy' cases with higher certainty). Context features are always words that occur in either the immediate surrounding of a name (local context) or in the document (global context). These words are mostly nouns since they are most informative. The similarity between the contexts of two names is then used to assess whether two names are co-referential or not. The following examples describe NE co-reference resolution methods that use various forms of context.

Examples are work from Bagga et al. (Bagga and Baldwin; 1998a) and work from Niu et al. (Niu, Li and Srihari; 2004) that builds on the work from Bagga. Both articles are about cross-document named entity co-reference resolution. The work from Bagga et al. describes a system which takes single co-referenced processed documents and tries to match names with all already seen names. This matching is not done using the names themselves but using the context of the names. In this case the context contains all the sentences that contain the name. These sentences are then represented as vectors and can be compared on common terms. If the similarity of two vectors is higher than a predefined threshold the names are concluded to co-refer. Bagga reports F-measures around 81%.

Niu et al. describe two observed problems using the above described method by Bagga. Firstly it is difficult to incorporate NLP results into the vector space model framework. Secondly the algorithm focuses on local pair-wise context similarity, neglecting global correlation in the data. In addition to usage of context of a name an information extraction module also extracts 1) other names within a predefined distance and 2) relationships associated with the name (leader-of, owner-of etc). These features are also used for name comparisons. This method is evaluated using a self made corpus (containing names with their feature sets) and compared with Bagga's method. Niu et al. present a F-measure of 88% in comparison with 64% accomplished by Bagga's method.

Peng et al. (Peng, He and Mao; 2006) use local and global context for the disambiguation of American location names. The local context consists of words directly before and after the name, global context consists of representative words for the text the name occurs in. The system extracts names from text and stores them together with their contexts in a profile. These profiles can be compared using the context words and popularity of a location (based on the population of the location). Peng et al. trained the weights for the three different scores on 17.755 newspaper articles. This training showed a large influence of local context and a smaller influence on the result by the global context. Final results of an evaluation on 300 documents show an accuracy of 78.8% (with a local window size of four words).

### 2.5.4 Using world knowledge

It is also possible to use external knowledge like dictionaries and encyclopaedia to solve co-referring names. Information extracted from these knowledge sources is used in addition to methods described above. World knowledge is often hard to use; extracting the right information is often difficult and can take a lot of computing time (parsing, scanning etc).
One example is research done by Bunescu et al. (Bunescu and Pasca; 2002). Their aim is to disambiguate names in a query using the other search terms in the query. Information extracted from Wikipedia is used to achieve this. They basically try to link the context words from the query to context words from articles about a person with the query name. They use the disambiguation page from Wikipedia to find all these different persons (and their articles). The system is trained and evaluated using examples extracted from Wikipedia. Results show an average increase of 10% on the accuracy of returned results.

## *2.6 Anaphora resolution*

Since NE co-reference resolution is related to anaphora resolution it is an obvious step to do a study of anaphora resolution research and common techniques. Most of the information and examples treated in this chapter were derived from Mitkov's 'Anaphora Resolution' (Mitkov; 2002) and from Jurafsky and Martin's 'Speech and Language Processing' (Jurafsky and Martin; 2000). This section will describe and assess the usefulness of some of the techniques used for anaphora resolution.

Anaphora are used to maintain cohesion in texts. When a part of a text refers (back) to some word(s) this part is called anaphor and the word group it refers to is called the antecedent. When both words have the same referent in the real world they are co-referential. If the antecedent is mentioned subsequently in the text the 'anaphor' is called a cataphor. Research by (Hobbs; 1978) found that 98% of pronoun antecedents were in the same sentence as the pronoun or in the previous one. Interesting for my research is the finding that it is not uncommon for proper names to refer to antecedents that are more than 30 sentences away!

The most widespread type of anaphora is that of pronominal anaphora (like he, him, its, their, herself, whom, whose, where etc.). However these are not necessarily anaphors. Verbs and adverbs can also be anaphors. For example: "Romeo begged for reinforcements, so did Dallaire". Sometimes the absence of a word (group) can also be anaphoric, this is called Zero Anaphora. An example is: "Jenny ordered thee copies of the document and Conny ordered several <zero anaphor> too".

Typically anaphora resolution consists of three steps: identification of anaphors, identification of possible antecedents and selection of an antecedent for an anaphor. The first two steps are usually not hard and can be done using lexical and syntactical methods. To minimize the search space the identification of possible antecedents for an anaphor is done within a certain scope (number of words or sentences) of the anaphor. The third step is harder and is more interesting with regard to this research.
There are a lot of different methods to select the right (best) antecedent depending on the amount of knowledge that can be used (semantic knowledge, world knowledge etc). Typically methods use constraints and preferences. Constraints are rules that must apply to the antecedent. Usually these are number and gender agreement with the anaphor. If there are more antecedents that fit the constraints preferences can be used to give an indication if an antecedent is 'good' or 'not good' for the given anaphor.

Different types of knowledge can be used to solve anaphora:

- Morphological or lexical knowledge: This knowledge concerns characteristics of words like gender and number.
- Syntactic knowledge: This is knowledge about the location and role of words and word groups in a sentence. Examples are Part-of-Speech tagging and grammars.
- Semantic knowledge: the knowledge about the meaning of word (groups) and if words can co-occur meaningfully.
- Discourse knowledge: knowledge about what a text or part of a text is about (the subject).
- Real world knowledge: this is knowledge about the world, usually extracted from specific knowledge sources.

Often a combination of these techniques is used to find the best antecedent. Different weights can be used to stress the importance of one technique over the other. Gender and number information are almost always used as constraints, the anaphor and the antecedent must have the same gender and number. A lot of times this is enough, only one antecedent is left within the search scope en thus this must be the right one. However this is not always the case and other knowledge must be used to select the best one from the remaining set of antecedents. Usually preferences are used to score the remaining antecedents. Throughout the years a great number of methods were used. Most of them are based on syntactic or discourse knowledge. Examples are the preference of certain syntactic roles (subject over object) and current 'focus' of the discourse over newly introduced possibilities. Pattern recognition in combination with a corpus and machine learning techniques can also be used to score antecedents.

Another important aspect of anaphora resolution is the way methods are evaluated. First of all there is a difference of evaluating a method on hand annotated data or on automatically processed data. Mitkov argues they should be done both in order to know how good the method strictly is and how good the method is in a system.
Mitkov proposes several forms of accuracy for the evaluation of anaphor-antecedent pairs. The normal accuracy is the percentage of correctly resolved anaphors. To distinguish between trivial (anaphors with only one possible antecedent) and hard cases also Non-trivial and critical accuracy are introduced. The first measures the number of solved anaphors with multiple possible antecedents. The second measures the number of solved anaphors with multiple possible antecedents after usage of number and gender constraints. These three scores give a good and complete insight into the true performance of these methods.

# 3 Novalink

## *3.1 Introduction*

The introduction already gave a short introduction into Novalink; a tool developed at TNO-ICT used to provide an overview of named entities that are related to the name the user is interested in. In order to do this Novalink has to find relations between NE's.
Novalink can be divided into two separate processes: a pre-process and a runtime process. Both processes are run independently and are described in the sections below. This research focus is to add named entity co-reference resolution to the pre-process in order to enhance the quality of the text mining done in the runtime-process (figure 2.1). The assumption is that information about co-referring names helps the runtime process finding more/better relations.

An important part of the Novalink system is the NNER which does the name recognition. These names form the basis for the NE co-reference resolution done in this research and hence it is important to know the mistakes the NNER makes. Section 3.4 describes the NNER evaluation done to assess the influence of the NNER on the NE co-reference resolution task. This section also describes the evaluation methodology, results and a conclusion.

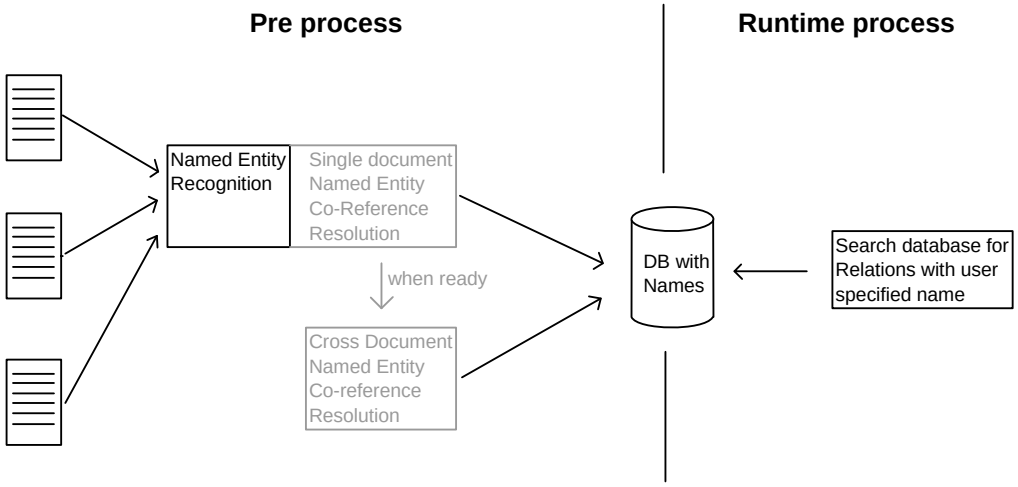



**Figure 2.1:** overview of Novalink (with the functionality added in this research in gray)

## *3.2 Pre-process*

Novalink's pre-process is responsible for filling a database with all the information needed for text mining. In short Novalink processes a set of news articles (in text form) to extract the names. These names and some meta-data extracted from the documents are put into a database. The named entity recognizer used to recognize the names is a machine learned recognizer. The NNER was trained on annotated text from the Belgian newspaper 'De Morgen' which was used for the CoNLL-03 language independent NE recognition task. Besides the annotation of names the texts were annotated with IOB and PoS tags (nouns, verbs, adjectives etc).
The NNER recognizes proper names. In some cases NNER also recognizes definite descriptions, mostly when a location name is used as an adjective (Dutch premier, Amsterdam's mayor etc).

Novalink uses a very simple rule to do NE co-reference resolution: names that are exactly the same are co-referential. This naïve rule causes several text mining problems; these are described in section 1.1.1.

## 3.3 Runtime-process

The actual text mining takes place in the runtime process and is initiated by a user by specifying a name. All the names in the database that match this term exactly are used to find relations. This is done in a statistical manner; co-occurrence of names is based on the number of documents both names occur in and the number of documents only one occurs in. The intuition behind this approach is that names that have a relation occur more frequently together than names that do not have a relation. The actual similarity is calculated using Dunning's Log-Likelihood (Dunning T.; 1993).
The top X strongest relations are displayed in a tree-like-graph as shown figure 1.1 in the introduction. Knowing exactly what names belong to what entity enables this text mining process to find relations with higher certainty and more results.

## 3.4 NNER Evaluation

### 3.4.1 Evaluation methodology

Since the goal is to evaluate the influence of the NNER on the co-reference task it was not enough to just count errors and calculate statistics. It must be clear what types of mistakes the NNER makes so the influence of these mistakes could be evaluated properly. There are a total of six different types of mistakes the NNER can make from which some can occur simultaneously in one annotation. The NNER:

1. missed a name
2. annotated a word that is not a name
3. split one name in multiple 'names'
4. concatenated multiple names into a single name
5. missed characters or word(s) that belonged to a name (incomplete)
6. annotated a few characters or words as belonging to a name while they do not (over complete)

The following table 3.1 shows examples of these mistakes as they were encountered during annotation of the corpus (annotations by are coloured yellow).

| | |
|---|---|
| **1 – Miss**<br>… maar dit is niet erg denkt Teeven. … | **2 - Not a name**<br>… Je bedoelt … |
| **3 - Split**<br>… Mink K. werd veroordeeld … | **4 - Concatenation**<br>… Bondscoach Otto Pfister van Togo … |
| **5 - Missed a word**<br>… Fred Teeven lacht … | **6 - Over complete**<br>… Wouter Leijnse is … |

**Table 3.1**: the six different mistakes made by the NNER

The Ground Truth data created for the evaluation of methods made in this research (described in chapter 5) was also used for the evaluation of the NNER. The data contains 271 annotated documents which were compared with documents annotated by the NNER. This comparison was done automatically using a very simple tool that recognizes and counts the different mistakes.
Most mistakes can be recognized very easily but some depend on pre-defined parameters. Classification of splits depends on the length of the gap between two annotations. If a gap is rather large the tool classifies the

mistake as 1 or 2. For this evaluation the gap size was set to 3 characters (two annotations closer to each other than the gap size are classified as a split). If the gap parameter was set larger mistakes 3 and 4 occurred more while mistakes 1 and 2 occurred less, the total number of mistakes stayed the same (within 0.5%).

### 3.4.2 Results

The following table shows the number of the separate mistakes.

| Mistake | Number | % mistakes | % of total |
|---|---|---|---|
| Missed a name | 217 | 11.8 % | 2.32 % |
| Annotated a name incorrectly | 566 | 30.7 % | 6.05 % |
| Split one name up in multiple names | 137 | 7.4 % | 1.46 % |
| Concatenated multiple names | 71 | 3.9 % | 0.76 % |
| Annotated a name incomplete | 503 | 27.3 % | 5.37 % |
| Annotated a name 'over complete' | 347 | 18.8 % | 3.71 % |
| | | | |
| incorrect | 1841 | - | 19.66 % |
| | | | |
| **Total** | 9363 | 100 % | |

**Table 3.2**: results of the NNER evaluation

As can be seen the 271 files contained a total of 9363 names from which 1841 were contain some kind of mistake. This means that the NNER has an accuracy of 80.34%. Note that this is not the amount of names the NNER recognizes; it is the percentage of names the NNER recognizes 100% correctly.
Recall and precision were calculated using the percentage of missing and incorrect recognitions so the NNER has a precision of 0.94 and a recall of 0.98. This shows that the NNER only misses very few names but has the tendency to annotate words that are no names.
Most of the mistakes made by the NNER are either the annotation of a normal word or incomplete annotation of a name (together 58% of all mistakes). Most of the words that were wrongly identified as a name are the first words in a sentence.
The NNER sometimes misses part of a name like the first- or last name. Mostly this is a random mistake but in some cases the NNER makes this mistake very consistent within a certain name in a text. It is unclear what part of the NNER causes this mistake.
As can be read in the table it is relatively rare that the NNER splits a single name up or concatenates multiple names.

### 3.4.3 Discussion of the results

Some of the mistakes made by the NNER can be explained by character-set conversion errors in the database of Novalink. Some of these errors resulted in the deletion of spaces and changes of punctuation marks. Errors like these can be the reason for some of the mistakes made by the NNER. For example the incorrect annotation of a normal word is sometimes the effect of the omission of a space character after a sentence ending. Usage of corrected documents will probably give slightly better results.
The tool used to count the mistakes is based on start- and end indices of words and not on the word strings them selves (to minimize implementation work). These indices are often shifted, meaning that the indices in the ground truth data vary slightly from the indices in the NNER-annotated data (the cause is unknown).

This makes it sometimes hard to distinguish between mistakes numbers 2 and 3 and between mistakes 1 and 4 (see the examples in table 3.3). So it is possible that the numbers of these mistakes are not totally correct. These situations do not occur that often so the evaluation accurately represents the quality of the NNER and the distribution of its mistakes.

| Annotation | Ground truth | argumentation |
|---|---|---|
| Henk van Zand<br>3 13 | Henk van Zand<br>0 9 | 'Henk van Zand' has a typical Dutch name form. But if the NNER misses 'van' and the name is split in half. But the evaluation tool might see 'Zand' as a separate name due to the higher index and erroneous classify this as mistake number 2 instead of 3. |
| Jan en Kees<br>3 10 | Jan en Kees<br>0 7 | In this case the opposite happens. 'Jan en Kees' are to separate names which should count as a 'concatenation mistake'. But instead the evaluation tool might think that the NNER missed a name (either Jan or Kees). |

**Table 3.3:** explanation of possible NNER-mistake-classification mistakes made by the evaluation tool. The start indexes of the important words in the examples are also given.

### 3.4.4 Conclusion

Even though the NNER makes a mistake once every five names the majority of these mistakes do not or very minimal change the surface form of a name. It is this surface form that is very informative and gives a first indication if two names are likely to co-refer or not. From this point of view the concatenation or splitting of name(s) has the biggest impact since they tend to change the surface form radically. Fortunately these types of mistakes are relatively rare (2.22% of all names).

The impact of incomplete or 'over-complete' names (mistakes 5 and 6) is hard to predict and largely depends on what has been annotated too much or too little. In general over-complete names will not be a big problem since the correct name is still contained in the annotation. Incomplete annotations on the other hand have lost a bit of information. This lack of information could make the comparison of names less accurate.

The third and last group of mistakes concern words that are either wrongly identified as a name or names that were completely missed by the NNER. Names that were missed simply do not exist for the system and hence they are not a problem for the co-reference task. Names that were erroneously identified will be compared with other names and can in that way introduce faulty data in the system. This is not so much a problem for name co-reference resolution but it is for the text mining task. However, this is no different from the current system.

Even though this evaluation was not meant to improve the quality of the NNER the results (and observations) indicate an area of improvement. It would help the NNER a lot if it would keep track of the recognized names and uses this information in a post process evaluating its own annotations.

In this post process it could compare recognized names with the other names (and their context) it recognized in the same document. Using this method it might be able recognize and correct splits, concatenations, over-complete and incomplete names. The idea behind this approach is the fact that a lot of names occur more than once in one document. Correctly recognized names can be used to identify and correct names that were recognized incorrect.

# 4 Evaluation methodology

## *4.1 Introduction*

Proper evaluation of the NE co-reference resolution methods is very important. Most important is of course the fact that the evaluation should give a good and clear indication of the performance and should point out the strong and weak points of the evaluated algorithm. This is not as trivial as it might seem, for different systems different aspects of the evaluation are important. For this research it is most important that the evaluation metric gives a good indication how good a methodology works and how much better (or worse) it works with respect to other methods. A set of requirements was used to select and refine a good evaluation methodology.

The evaluation metric should:

1. Give a good indication of the performance of the evaluated method. This means:
   - Mistakes made must be counted the same (regardless of the group of names the mistake was made in)
2. Give a good indication of the strong and weak points of the evaluated method:
   - indicate if names are erroneously grouped together vs. names that are erroneously left out of a group
3. Give a clear indication how much better or worse one method is compared to another
4. Be understandable for other computer scientists active in related research-fields

First a number of evaluation methods proposed by literature are 'evaluated' with respect to the requirements. Secondly it is described how one of these methods is adopted to suit the requirements. The method selected is used for both single- and cross document named entity co-reference resolution evaluation.

## *4.2 Commonly used evaluation metrics*

There are a lot of research fields that that have high similarity to this research and hence evaluation metrics from these fields might be very usable. Fields closely related are Information Retrieval, Anaphora Resolution and Clustering. Evaluation metrics commonly used within all three fields are precision and recall (combined within an F-measure) which originate from the field of Information Retrieval.
Precision is the portion of correct answers that was truly correct. Recall is the portion of the correct answers found by the method with respect to the total number of correct answers. Search engine results are commonly used to explain these two concepts. If for example a search engine returns 10 items from which 5 are correct the precision is 5/10. If there are 25 correct items in total (on the entire internet) the recall is 5/25. In general methods that maximize precision tend to have a low recall score (or the other way around). If for example a search engine would only return 1 item that is correct the precision is 1.0. But at the same time the recall is 1/25. The other way around is also possible: return all the items on the internet, getting a recall of 1.0 but a precision of almost 0.0.

Precision and recall can be combined in a single score called the F-measure (weighted average of precision and recall). Within F-score it is possible to give preference to precision over recall (or the other way around). Precision and recall are also used for the evaluation of most MUC tasks. In general these are good performance measures and also give insight in the strong and weak points of the evaluated method.

Because of possible inconsistent usage of recall and precision measurements for anaphora resolution, Mitkov (Mitkov; 2002) proposes 'success rate'. The success rate simply is the number of correct answers divided by the total number of answers. Even though it is a relatively good performance indicator it does not give a lot of insight into the evaluated method. This metric only indicates how many of the names are correctly 'paired up', but it does not really give information what causes a specific success rate (are names erroneously clustered or not). Because of this it does not satisfy requirements one and two.

Another evaluation metric used within the field of clustering is Ctrk (Yang, Yoo, Zhang and Kisiel; 2005). For every cluster a Ctrk score is calculated and these scores are averaged for a single score. Ctrk is a cost function that lets the user set the costs for misses (documents that should have been in a cluster) and false alarms (documents that should not be in a cluster). These costs are normalized using the observed chance on a mistake and the chance that a mistake occurs at random within the data set. In the end the lower the cost the better the evaluated method performs. This evaluation metric gives a good view on the overall performance. Even though the final score is not very informative with respect to the strong and weak points of the evaluated method the score is made up from other scores that do give an insight. Each individual score tells something about the 'hardness' of a certain cluster, in this case a certain name. Clusters and scores can easily be evaluated by a human to see what names turn out to be hard and why (depending on cluster size).

Two of the above described evaluation metrics (Ctrk and precision/recall) are largely the same with respect to the given requirements. The Ctrk does not give a lot of insight on its own but need to be evaluated to be useful. Precision and recall do not have this drawback and are frequently used. Hence precision and recall are used for the evaluations in this research.

## 4.3 Adaptation of a Metric for Named Entity Co-reference Resolution

With the selection of precision, recall and F-measure as evaluation measures it is yet unclear what these terms exactly mean with respect to NE co-reference resolution or how to calculate them properly. To answer these questions the terms that make up precision and recall should be defined properly.

The common functions used to calculate precision and recall are:

$$precision = \frac{\#true\ positive}{\#true\ positive + \#\ false\ positive}$$

$$recall = \frac{\#true\ positive}{\#true\ positive + \#\ false\ negative}$$

- True positive: the number of names in a co-reference group that really co-refer.
- False positive: the number of names in a co-reference group that do not co-refer with the other names in that group
- False negative: the number of names that co-refer with other names in a co-reference group but are not a member of this group.
- True negative: number of names that are not co-referential and are not members of the same group (not used for precision and recall calculations, but added to give a complete list).

The data that needs to be evaluated has groups (clusters) of names that are determined to co-refer to the same entity. So names that were clustered together by the algorithm that were also clustered by human annotators were solved correctly and hence are true-positive. If a cluster defined by the algorithm misses names then these names were wrongly classified as not belonging to that cluster and hence are false-negative. It is also possible for an algorithm to cluster together to many names and thus the names that should not be in the cluster are false-positive.

Even with these definitions it is not a clear cut how to calculate precision and recall per cluster, nor how to determine the overall precision and recall (per document or per document set). The following sections describe methods created (based on evaluation methods in other research fields) and methods found in literature to do this.

### 4.3.1 MUC co-reference resolution evaluation scheme

The evaluation method proposed for the MUC cross-document NE co-reference resolution task was proposed by Vilain et al. (Vilain, Burger, Aberdeen, Connolly and Hirschman; 1995). This evaluation method is based on the links needed to make a minimum-spanning tree connecting the names that are co-referential. The links in the tree are used to calculate recall and precision scores. The example given by Vilain et al. has ground truth data consisting of {A-B, B-C, C-D} which is compared with {A-B, C-D}. The recall score is 2/3 because 2 of the 3 links are found, the precision is 2/2 because both links are correct (see figure 4.1).

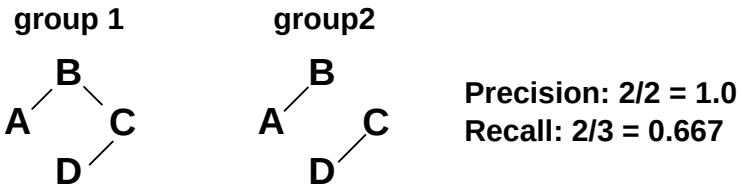



**Figure 4.1**: Group 2 has 2 edges, from which 2 also occur in group 1; hence the precision = 1. But group two only has 2 from the 3 links in group 1 so the recall is 2/3.

In my opinion this method has two drawbacks:

- Precision and recall are calculated using the links between names instead of the names themselves. In order to give accurate results it should be about the names grouped together. In my opinion a recall score of 2/4 gives a more accurate indication of the situation than 2/3 since both {A-B} and {C-D} have 2 from the 4 co-referential names and not 2/3 because of the links.
- The method does not distinguish between mistakes made in large groups and small groups. In the end it is about the 'missing links' and this is independent of the size of the sets where these links are missing. Arguably this is undesirable because mistakes can have more or less impact on the total co-reference task depending where the mistake took place. So this method fails requirement 1.

### 4.3.2 Name matching approach

The next is based on name pairs in the same way anaphora resolution evaluates anaphor – antecedent pairs. Every possible name pair in the system is evaluated. For each pair of names (pair1) in the automatically annotated data the corresponding names (pair2) in the ground truth data are identified. From pair 2 it is checked if they are co-referential or not and then pair 1 is evaluated using this knowledge:

- If both pair 1 and pair 2 are co-referential pair 1 is counted as true-positive.
- If pair 1 is found to be co-referential but pair 2 is not, pair 1 is counted as a false-positive
- If pair 2 is co-referential but pair 1 is not, pair 1 is counted as false-negative
- If both pairs are not co-referential count pair 1 as true-negative

In this way a confusion matrix can be created and precision and recall can be calculated using this matrix. For the evaluation of multiple documents the average of the different scores for each document can be used.

A drawback of this method is the large number of comparisons (n*(n-1)/2). For a document with only 10 names 45 name pairs are evaluated, resulting in a confusion matrix with a total sum of 45. The vast majority of comparisons will be true-negative since most word pairs do not co-refer. Because of this, accuracy and success rate measures will not be representative. However recall and precision will be representative since the number of true-negatives is not used in calculation of either of them.

Another drawback is that errors in large clusters have a lot more impact on the evaluation than errors in small clusters. If one name is erroneously left out of a cluster with five other names the false-negative field will be raised five(!) times lowering recall score drastically. If the same thing happens to a cluster containing only two names the false-negative field will be raised only once. Even though the mistake is the same the penalty will be larger if the cluster size the mistake occurred in was larger. An example is given in table 4.2.

| # | Scenario | Result |
|---|---|---|
| 1 | Exact same set:<br>[jan, jan, jan]<br>[piet, piet]<br>[kees] | 4 \| 0<br>0 \| 11<br>Precision: 1.0<br>Recall:1.0 |
| 2 | One 'jan' is classified as a separate name:<br>[jan, jan]<br>[**jan**]<br>[piet, piet]<br>[kees] | 2 \| 0<br>2 \| 11<br>Precision: 1.0<br>Recall:0.5 |
| 3 | One 'jan' is classified as a 'piet':<br>[jan, jan]<br>[piet, piet, **jan**]<br>[kees] | 2 \| 2<br>2 \| 9<br>Precision: 0.5<br>Recall:0.5 |
| 4 | One 'piet' is classified as a separate name:<br>[jan, jan, jan]<br>[piet]<br>[**piet**]<br>[kees] | 3 \| 0<br>1 \| 11<br>Precision: 1.0<br>Recall:0.667 |

**Table 4.2:** In every example a set of name-clusters is compared to this ground truth set: {jan, jan, jan}, {piet, piet}, {kees}

The example above shows the impact of one error in different situations. Firstly it is shown that a simple mistake can result in precision (and recall) of 0.5! Secondly, the exact same mistake occurring in different clusters results in different scores (example 2 and 4).

This inconsistent rating of errors makes this method unsuitable for calculation of precision and recall (it fails requirement 1).

### 4.3.3 Clustering matching approach

The second approach compares groups (clusters) of names (not pairs of names) analogue to clustering evaluation methods. Each cluster in the ground truth data is matched with a cluster in the automatically generated data. These clusters are compared and for each cluster precision and recall scores are be calculated.

The first step is to match a cluster from the ground truth set with a cluster from the evaluation set. Clusters are not always easy to match since clusters in the evaluation set can differ (a lot) from clusters in the ground truth data. Clusters can contain a different number of names, can be split or concatenated together. Some clusters in the ground truth set might not exist in the evaluation set (or the other way around). There are multiple criteria to compare clusters:

- total number of names two clusters have in common
- recall between two clusters
- precision between to clusters
- F-measure of two clusters

The most intuitive method is to simply match clusters on the number of names they have in common. Unfortunately this matching criterion favours recall over precision and it is arguable that it does not always select the best matching cluster (see figure 4.3).
An alternative proposed by Agarwal (Agarwal; 1995) uses F-measure to match clusters. This method selects clusters based on both precision and recall and does not favour one of them over the other (unless weighted F-measure is used). An example (figure 4.3) using the above described method is described below. Looking at this example it is evident that matching based on F-measure is more 'honest' than matching based on common names.

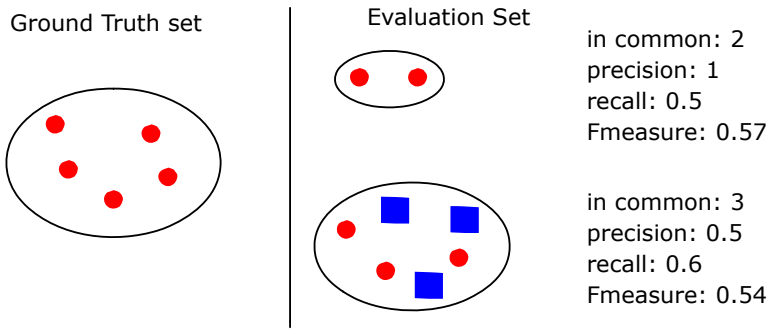



**Figure 4.3**: a cluster from the reference set has two matching clusters. The possible matching 'criteria' are stated beside the clusters on the right side. Based on the number of names the clusters have in common cluster number two matches best. However, this cluster contains 3 other names lowering precision and F-measure

Using a method to match clusters and calculate precision and recall for each pair of clusters does not give a performance score for a single document. The most natural way is to average the scores for the individual clusters into single precision and recall scores. However, simply dividing the sum of the individual scores by the total number of clusters does not give a representative score. The reason is that small clusters contribute the same to the final score as large clusters while these have a higher probability of having lower scores (fails requirement 1). In this way document scores are higher than they should be. A solution is to multiply each cluster-score by the number of names in the cluster and in the end divide the sum of these scores by the total number of names data. In this way larger clusters contribute more to the final precision and recall scores.

A short preliminary study of the ground truth data showed that 2/3 of the clusters only has one name. Even though the final scores are normalized using the number of names within clusters the great number of one-name-clusters still has a big impact. Especially since the chance on mistakes in these small clusters is relatively low they are likely to raise the precision and recall scores. Hence not evaluating clusters with only one name will give more accurate scores. Mistakes made with these one-name-clusters will still be incorporated in the result since they have impact on longer clusters that are evaluated. In this way the disproportional influence of small clusters on the final scores is reduced while mistakes made within these clusters are still evaluated. This is the same as the notion of 'critical' anaphora proposed by Mitkov. Anaphora that are simple to solve can be left out of the evaluations leaving only the hard (critical) cases.

Independent of the matching method used, there is always the possibility that clusters in either the ground truth data or evaluation set are not matched due to erroneously recognized names. Same as with the 'name matching approach' it is hard to predict how these clusters influence precision and recall. As proposed before, these mistakes should be measured separately.

The way to calculate precision and recall based on clusters is basically the same as the B-CUBED method proposed by Bagga et al. (Bagga and Baldwin; 1998a) and (Bagga and Baldwin; 1998b). The main commonality (and difference with the MUC method) is that the scores are weighted using the cluster size. In this way the drawbacks of the MUC method as described at the end of section 4.3.1 is omitted. Still the method described by Bagga et al. differs a bit from the proposed method. Firstly it does not describe the matching method(s) used to select clusters used to calculate precision and recall with. Secondly they do not describe special treatment of one-name-clusters.

## *4.4 Summary*

A number of different evaluation metrics were described and evaluated using the requirements stated in section 4.1. The method that fulfilled all the requirements compares clusters of names in the ground truth data with clusters of names created by a co-reference resolution method. For each cluster of co-referential names in the ground truth data (called GT) the following is done:

1. Find the best matching cluster of names created by the co-reference method using the F-measure as criterion (this group is called CM).
2. Calculate the precision and recall scores of CM with respect to GT (using the members of the two clusters).
   - Precision: number of correct names in CM / total number of names in CM
   - Recall: number of correct names in CM / total number of names in GT
3. These scores are weighted using the number of names in GT, groups with only 1 name are not used in the evaluation.
4. The final score is the sum of all the weighted scores divided by the total number of names in the ground truth data

There is a large probability that not all of the clusters found by co-reference resolution methods will be evaluated simply because they do not match a clusters in the ground truth data. These clusters will simply be counted separately and will be called 'over complete clusters'.
The opposite is also possible, clusters in the ground truth data that do not correspond with one cluster found by co-reference method (mostly due to NNER mistakes). These clusters will also be counted separately and will be called 'incomplete clusters'.

# 5 Corpus creation

## 5.1 Introduction

The selected evaluation method described in chapter 4 compares co-referential names found by a method with co-referential names in the ground truth data. Unfortunately the correct data needed was not available (texts with annotated Dutch named entity co-references) and hence needed to be created.
This chapter describes the creation of a corpus that was used to evaluate both single- and cross-document co-reference resolution methods (described in chapters 7 and 8). This means that the ground truth data (in form of an annotated corpus) should contain cross-document named entity co-reference annotations. This makes the creation of such a corpus very hard; somehow it should be clear what names in the entire corpus are co-referential. Such a corpus did not yet exist for Dutch and hence it was needed to be built from scratch.
For evaluation purposes it was decided that the corpus should resemble the entire data set as closely as possible. The reason behind this decision was that the creation of a more general 'named entity cross-document co-reference' corpus would differ too much from annotations created by the NNER. That would make the corpus unsuitable for evaluation in this research.
To be able to train/tune and evaluate different methods on all the possible problems the corpus has to contain these problems:

- Similar names that refer to different NE's
- Different names that refer to the same NE
- Definite descriptions (which are often non-rigid designators)

The following set of requirements was used for the creation of this corpus:

1. The corpus should contain cross-document named entity co-reference annotations.
2. The corpus should resemble the data as closely as possible
3. The corpus should contain the 'problem names' described above.

First the selection of documents, annotation rules and attributes are described. Secondly it is described what software the annotators used and how they where aided. Finally the process is evaluated presenting different kinds of observed problems, inter-annotator agreement and post-processing of the corpus.

## 5.2 Document selection

The first thing that had to be determined was the size of the corpus. The corpus needed to be big enough to fulfil requirements two and three but it should be as small as possible to minimize annotation work. A corpus with a size of 300 documents was found good enough to fulfil these two demands. The aim of the corpus is to resemble the entire data set as good as possible. One way to do this is to select random documents. However, with a corpus size of only 300 documents it is very likely that names do not occur in a lot of documents, or only in one document. This is not a representative sub-set of the complete data set with thousands of documents containing the same name. To cope with this problem a set of six categories made up out of 50 documents each were created.

All the documents in one category share at least one name (maybe more). This name does not necessarily belong to only one entity. Two of the six categories are filled with documents containing a very general name (Jansen, Oranje) resulting in documents about a lot of different topics.
The six categories are:

- Holleeder
- Kuijt / Kuyt (2 spellings of the same name!)
- Angels
- Fortuijn / Fortuyn
- Jansen
- Oranje

Some of these categories represent different NE co-reference resolution problems. The Kuijt category has documents with two different spellings of the same name. The Fortuijn category is special since Pim Fortuijn deceased in 2002 but his name still exists referring to multiple different things (person and organization). The Jansen and Oranje categories have very common names referring to different things, this in contrast with the Holleeder and Angel's categories that mostly have names referring to the same entity. Naturally there are a lot of other names in all the documents and the exact number of names and content is unknown. However the same goes for the real data.

It is very likely there will be names (other than the category names) that exist in documents in multiple categories. However, to make sure categories are not independent sets without any relations with other categories a set of documents containing two 'category-names' (like Holleeder and Angels) were introduced into the corpus. In this way the categories Holleeder and Angels are linked by 22 documents and Fortuijn is linked with Jansen using 5 documents.

In this way the corpus fulfils requirements 2 and 3.

## *5.3 Annotation and attributes selection*

The NLP framework GATE (General Architecture for Text Engineering) was used to do the annotations (Cunningham, Maynard, Bontcheva, Tablan; 2002). This tool enables users to annotate text and assign attribute-value pairs to the annotation. In this annotation work named entities had two attributes:

- *Chain:* this attribute holds the name of the entity the NE is referring to (Holleeder might have as chain-value 'Willem Holleeder'). The aim is of course to have the same chain-value for all the names that refer to the same entity in the world throughout the entire corpus (for cross document co-reference resolution). This was of course a difficult task, especially with different annotators. How this problem was minimized and solved in a post process will be described in section 5.7.
- *Type*: this attribute refers to the category of the named entity. This attribute does not occur in the requirements but was added to be able to assess the usefulness of NE type information for this research (one of the research questions). There are five possible types: Person, Organization, Location, Event and Other. The first three are used for Automatic Content Extraction (ACE) as defined by the Linguistic Data Consortium (Linguistic Data Consortium; 2005)). The last two types (Event and Other) are not described in the ACE annotation guide. The event type was introduced after a pilot of 50 documents which showed a lot of events described in news articles.

More formal descriptions of the types (all but the last originate from the ACE annotation guide):

- Person - Person entities are limited to humans. A person may be a single individual or a group.
- Organization - Organization entities are limited to corporations, agencies, and other groups of people defined by an established organizational structure.
- Location - Location entities are limited to geographical entities such as geographical areas and landmasses, bodies of water, and geological formations.
- Event – temporal and/or periodical events like World Championships, birthdays, elections etc.

During the initial study it was found that these categories cover most of the NE's encountered in the documents. Persons, organizations and locations are very basic categories for names in texts. The event type occurred relatively often in news articles, more than in normal texts. This can be explained by the simple fact that news often describes events (like sport-event, political-event or a random event).
Other categories in the ACE set (GPE, weapons, facilities and vehicles) were very rare and not worth using.

## *5.4 Annotation aiding*

Annotating text is a time-consuming process where a lot of choices depend on interpretation. To aid the annotators and hence achieve higher inter-annotator agreement the entire corpus was automatically annotated using the NNER. This recognizer annotated NE's and tried to guess the chain-values. In this way annotators did not have to annotate all NE's 'from scratch' but only had to check existing annotations, changing errors and setting attributes where necessary. In this way the two most interpretation-sensitive annotation-aspects (what is a NE exactly and the chain attribute) were guided. There are still a lot of cases the NNER made a mistake that needed to be corrected but these mistakes are mostly deterministic (occurring the exact same way multiple times). Unfortunately the 'freedom' in which names can be used within a text is rather large. This in combination with faulty recognitions made by the NNER makes some annotations really tricky. The documents belonging to one category were assigned to the same annotator, in this way was easier to annotate cross-document NE co-references.
The annotators were also provided with an annotation guide describing how GATE works and how (and what) to annotate. This annotation guide can be found in Appendix A (it is in Dutch).

## *5.5 Observed problems*

This section describes some problematic named entities encountered by the annotators. This is meant as an indication how hard annotation could be and to show the difficulty of the overall problem. For most of the examples shown below there were no straight forward annotation rules and they hence were not annotated in a single unique way. The underlined parts were identified by the NNER.

- Johan 'de Hakkelaar' Verhoek: The name of the person is split in half and a nickname is inserted in the middle. The NNER distinguished three names. The best way to annotate this would be to annotate the entire phrase as one name or split 'Johan Verhoek' and 'de Hakkelaar' and give them the same chain value. Unfortunately the last option is not possible and hence the entire 'phrase' was annotated.

- Belgische tak van de Hell's Angels: In this case 'Belgische' is an adjective telling something about the Hell's Angels. As in the previous example it is not possible to annotate the entire phrase simply because the other words do not belong to the name. Hell's Angels can still be annotated with 'Belgische Hells Angels' as its chain value but it is unclear how to annotate the single word 'Belgische'. The most natural solution is to simply annotate Belgische with a chain value of 'Belgium' because it refers to that country.
- Lebbis en Jansen: This case is highly ambiguous since 'lebbis en Jansen' refers to a group of people but it can also refer to two person individually ('Lebbis' en 'Jansen'). Using world knowledge and context the first option was more likely but this would be a very hard case for a machine.

## *5.6 Inter annotator agreement*

As described by Mitkov (Mitkov; 2002) it is preferable that each document is annotated by at least two annotators. Unfortunately that was far from feasible for this project. In total four people annotated a portion of the corpus. None of the annotators had a lot of experience with annotating texts.

To at least be able to get an indication of annotator agreement and to have the possibility to correct annotations set of five documents were selected and given to all the annotators. The set of documents were selected from 5 different categories, all with an average length and a diversity of names and types. The annotators did knot know about these documents (although they might have seen strange compared to the rest of the category they had to annotate). From these documents the number of names belonging to the same cluster can be counted and used to calculate inter annotator agreement. However, this is more an indication of the difficulty of the annotation-task rather than the quality of the corpus since all the documents were post-processed further to get a more consistent corpus.

Precision, recall and F-measure scores were used to calculate the 'inter annotator agreement' instead of Cohens Kappa (Cohen; 1968). The reason is that in order to calculate Kappa a confusion matrix (also called contingency table) is needed. But since the evaluation methodology does not calculate the number of true-negatives it was not possible to use the Kappa score.

The evaluation method as described in chapter 4 is used to compare the five documents annotated by every annotator. This method calculates precision, recall and F-measure. Table 5.1 shows the F-measures between the different annotators.

| | Annotator 1 | Annotator 2 | Annotator 3 | Annotator 4 |
|---|---|---|---|---|
| Annotator 1 | 1 | 0.92 | 0.97 | 0.97 |
| Annotator 2 | - | 1 | 0.93 | 0.91 |
| Annotator 3 | - | - | 1 | 0.99 |
| Annotator 4 | - | - | - | 1 |

**Table 5.1**: inter annotator agreement results (F-measure)

The recall and precision scores are not stated here but the precision between annotators is very high (0.99 on average, between some annotators even 1). This means that the recall is much lower (0.90 on average). This can be explained by the fact that there are names referring to multiple possible entities from which annotators seem to have a hard time choosing the same. For example nation names can be very hard, does a nation name refer to the inhabitants of the nation or to its government?

Apparently it is sometimes very hard to point out the right entity a name is referring to. Maybe there is not always a single unique entity a name refers to. This can indicate that the strict annotation of this corpus is a bit to strict and not totally realistic (to have names referring to different aspects of one NE).

## *5.7 Post-processing of the Corpus*

For this corpus it is important that every entity has the same 'chain-value'. Unfortunately it is very likely that different annotators gave the same entity (slightly) different chain-values. It is important that these chain-values are correct throughout the entire corpus and hence a post process is needed to find and correct any inconsistencies.

A small tool was used to identify chain-values that could denote the same entity. A human operator was then prompted how to deal with this problem. As a last check all the documents were parsed and the unique chain values were loaded into an Excel sheet. These chain names were sorted and then check manually. In this way the last inconsistencies were corrected resulting in a corpus with cross-document chain values.

# 6 Single-document named entity co-reference resolution

## *6.1 Introduction*

The fact that names exist within the same document is a very strong piece of information with respect to the NE co-reference resolution task. It gives an indication that NE's that have a high lexical similarity (i.e. look the same) point to the same entity in the world. This notion is also described in research as the "single sense per discourse" principle first defined by Gale et al. (Gale, Church and Yarowsky; 1992). Even though this does not directly translate to "one NE-sense per document" is has been successfully used in this way in for example location disambiguation by Li et al. (Li, Srihari, Niu and Li; 2003). David Yarowsky (Yarowsky; 1995) showed that this principle was highly consistent regarding names. During annotation work (described in chapter 5) this principle was found to be true most of the time. Looking at news articles it is not likely that two persons with the same name are discussed in the same text. Even if this would be the case most writers make sure that readers know at any point in the text what person a name refers to. This is not always done by using full names because readers also understand what ambiguous names refer to using the context. Other names that do not refer to persons can be more ambiguous. Especially names of countries can refer to a lot of different things (its people, its government, its military etc). These are concepts that are rarely properly introduced in the text but are explained by context and by usage of common sense.
Still, the fact that names occur in the same text conveys useful information. Hence it is logical to start the NE co-reference resolution task on document level.

Basically this methodology is based on the hypothesis that:

> *Names that have a high surface form similarity and occur in the same single document are co-referential.*

Multiple methods were developed, evaluated and compared using this hypothesis. To be able to do this comparison a clear criterion must be defined. The following criteria were used:

1. The method must have (or maintain) a high precision score
2. The method must have a high recall score

The reason to use these criteria is two fold:

1. The final goal is to improve the overall NE co-reference resolution (following the assumption that this improves the text mining process). Since the existing Novalink method already achieves very high precision the improvement must come from recall (described in 6.2).
2. It is important that the methods realize a high precision since the groups of co-referential clusters of names are used for cross-document name resolution. If there are 'a lot' of mistakes in these clusters to start with it will be hard for any cross-document method to work properly.

The goal is slightly contradictory because methods that raise recall tend to lower the precision; hence a suitable balance needed to be found. Since the effect of NE co-reference resolution on text mining was yet unknown the (balanced) F-measure scores were used to decide which of two methods is

better. In case of doubt the solution with a higher precision score (because of reason 2) is defined to be 'better'.

This chapter describes the study done to use the hypothesis properly and assess whether it is a valid and useful assumption to develop good single document NE co-reference resolution methods using the criteria described above. To do this, first a baseline method is introduced to provide a minimal performance level for this study. More sophisticated methods are defined using the results from this baseline method and techniques proposed in literature. The evaluation of these methods is used to conclude whether the hypothesis is valid or not.

## *6.2 Baseline string-comparison method*

To be able to evaluate the quality of 'single document NE co-reference resolution' methods a baseline method should be created as a reference point. Using this baseline method the (extra) quality of more elaborate methods can be evaluated with respect to the computational complexity. This is important because "it may not be worth while developing a specific approach unless it demonstrates clear superiority over simple baseline models" Mitkov (Mitkov; 2002).
Some kind of string comparison should always be done as an indication whether two names are the same name or not. Especially in the case of single documents where similar names are very likely to refer to the same entity, a string-distance based method might perform well. Logically two totally different names that point to the same entity will not be found by this kind of method. On the other hand names that have the same surface form but do not co-refer will be erroneously matched together. However, the methods used for single-document NE co-reference resolution work with the assumption that this does not (or rarely) occur within one document.

### 6.2.1 Implementation of baseline method

Currently Novalink uses a very strict (case sensitive) comparison method to find 'NE co-references'. This is a very good baseline method for this research because:

- It is simple and easy to implement
- It makes it possible to compare other methods with the way Novalink currently handles co-referential names (in contrast with a slightly different baseline).

The rule used to conclude whether two names are exactly the same is the following:

> *Two names (in the same document) are co-referential if they are spelled exactly the same.*

### 6.2.2 Evaluation of the baseline method

To properly evaluate the NE co-reference resolution methods described in this entire chapter they should be evaluated using the ground truth data and the evaluation methodology described in chapter 4. The baseline method was used to find co-referential names in both human and NNER annotated documents. The first evaluation will give a clear view on the true performance of the method while the second evaluation will give an indication how the method would work within a real system using imperfect data. A set of 70 (25% of total) documents, randomly extracted from the corpus that were not used for testing and tuning was used for the evaluation.

**Results**

The following table (6.1) shows the average precision, recall and F-measure scores for the 70 evaluated documents (for both ground truth and NNER annotated sets). The explanation of these results can be found in the discussion section (6.2.3). Bear in mind that these results are calculated using only groups of names with more than one name in it.

| Evaluation metric | Human annotated names | NNER annotated names |
|---|---|---|
| Precision | 0.99 | 0.99 |
| Recall | 0.73 | 0.65 |
| F-measure | 0.84 | 0.78 |
| Fully correct clusters | 37% | 26% |
| Incomplete clusters | 0 | 32 |
| Over complete clusters | 0 | 102 |

**Table 6.1**: results of the single-document NE co-reference resolution Baseline method

These results show extremely high precision and relatively low recall scores. Since the precision is so high the small number of 'fully correct chains' can only be explained by the low recall. Apparently a lot of clusters miss names.
It seems that mistakes made by the NNER cause lower recall scores because these mistakes tend to 'alter' the surface form of names and hence are not compared correctly.
As described in chapter 6 the numbers of clusters missed (incomplete) and annotated too much (over complete) are counted separately. These are mostly the result of mistakes made by the NNER and their distribution is largely the same as observed during the evaluation of the recognizer. Most of these clusters (~90%) only contain one name.

### 6.2.3 Discussion of the results

The biggest problem of the baseline method is the recall; not all names that should be clustered are grouped together. The cause of this problem is the strict way names are compared. Research by Nenkova et al. indicates that names tend to evolve throughout texts (Nenkova and McKeown; 2003). They claim that of all initially used names 76% is changed in subsequent mentions of the same entity. This roughly concords with the percentages of fully correct chains.
Looking at the 'misclassified' names there are a few common similarities (ordered by importance):

- Only part of the original name is used (only first name or last name instead of the complete name)
- Abbreviations are used, commonly organization names, or initials of person names
- A nickname or totally a different name is used to refer to the same entity.

In order to cope with these problems the comparison method used must be more flexible.

The precision is very high but not perfect. This can be explained by a small number of names that are exactly the same but are not co-referential. These names are not very common in the corpus but there are some:

- Willem: a very 'common' name with the documents about the Hells Angels and Willem Holleeder (Willem van Boxtel, Willem Endstra, Willem Holleeder). So Willem on its own can refer to multiple NE's so it is easy for the baseline method to make a mistake.

- Nederland: depending on the context 'Nederland' can refer to multiple things and thus two 'Nederland' occurrences in one text don't have to be co-referential. This potentially occurs in all the categories but is more prominent in the 'oranje' category.

The high precision scores also indicate the validity of the hypothesis; names that are exactly the same and occur within the same document are co-referential. However, the way the hypothesis is used in the baseline method also causes lower recall scores.

## *6.3 Advanced method*

A first step to make a better single document named entity co-reference resolution method is to improve the baseline method by overcoming the problems of the baseline. The main problem identified was the strict comparison method used which causes lower recall scores. This chapter describes a more flexible comparison method that tries to handle the problems described above (mostly focused on problem one). The final goal of this method is to raise the recall with a minimum effect on the precision score.

### 6.3.1 Selection and implementation of advanced method

The most important point of failure of comparison method is that it compares entire names, even when the names consist of multiple words. A solution is to compare these individual words rather than the complete names. A simple rule can be used to do this:

> *If one of the words within two names match exactly they are co-referential.*

An obvious pitfall for this rule is names that share a common word (like 'van', 'de' in Dutch). Normalization can be done to avoid these kinds of mistakes (Branting; 2003). In this method two normalization rules were added:

- Remove words that are common in Dutch names ('van', 'de', 'der').
- Remove any punctuation marks located in a name (to avoid mistakes made by the NNER).

In the baseline method all the names in a cluster are exactly the same so a new name needed to be compared with only one member of the clusters. With this method this is no longer the case, clusters of co-referential names can consist of different names. So every name must be compared with the already existing clusters in a meaningful way and should be either added to a cluster or be a new cluster on its own. There must be some kind of threshold that can be used to determine this.
There are multiple ways how a name can be compared with a set of names:

1. Compare the new name to all the names in the cluster and select the highest score.
2. Compare the new name to all the names in the set and average the similarity scores.

These two options were both evaluated in a small scale test. In this test 75 names were compared with 10 clusters (clusters containing 2, 3 or 4) names. Half of the names were 'positive' meaning that they should match one of the clusters.
The second method had a higher accuracy than the first method. The first option has as the disadvantage that names can incorrectly be added to a cluster because they matched very well with only one member in the set.

During the test the following cluster was made using this method: [oranje, willem van oranje, FC willem II] where three totally different names were put in the same cluster.
The first method scored the best with a threshold of 0.85 (similarity score > 0.85 then the name belongs to the cluster, else it does not). Hence this option was used for the baseline method.

An example of this method: similarity between the name 'Jan Peters' and the cluster {Peters, Jan M. Peters}. First the similarity with the individual members of the cluster is calculated (see table 6.2). The similarity of 'Jan Peters' and Peters is 1.0 (1/1). The similarity of 'Jan Peters' and 'Jan M. Peters' is 1.0 (2.0/2). The final similarity of the given name and the cluster is (1.0+1.0)/2 = 1.

| **Words of name1** | **Words of the cluster** | **Similarity Score** |
|---|---|---|
| Jan | Peters | 0.0 |
| Peters | Peters | 1.0 |
| | | |
| Jan | Jan | 1.0 |
| Jan | M | 0.0 |
| Jan | Peters | 0.0 |
| Peters | Jan | 0.0 |
| Peters | M | 0.0 |
| Peters | Peters | 1.0 |

**Table 6.2**: the similarity scores between 'Jan Peters' and the members of the cluster {Peters, Jan M. Peters}

This method simply compares all the words in two names without using any knowledge about names. Using this methodology the first word from a name will be compared with the last word from the other name even though it is very unlikely that they match. The knowledge of the internal structure from names can be very helpful. Unfortunately this knowledge was not available (information about name category is also needed).

### 6.3.2 Evaluation of the advanced method

The evaluation of the advanced method was done in the exact manner as the evaluation of the baseline method. Also, the exact same files were used.

**Results**

The following table (6.3) shows the average precision, recall and F-measure scores for the 70 evaluated documents (for both ground truth and NNER annotated sets). Abbreviations are used as column names to save space in order to include the baseline results. The 'human annotated names' will be abbreviated with HAN and the 'NNER annotated names' with NAN. The results of the baseline method will have a B as first character (BHAN, BNAN) and the results from the advanced method will have an A as first letter. The results of the advanced method are shaded grey.
The explanation of the results can be found in the discussion section (6.3.3).

| Evaluation metric | BHAN | BNAN | AHAN | ANAN |
|---|---|---|---|---|
| Precision | 0.99 | 0.99 | 0.95 | 0.91 |
| Recall | 0.73 | 0.65 | 0.88 | 0.82 |
| F-measure | 0.84 | 0.78 | 0.91 | 0.86 |
| Fully correct clusters | 37% | 26% | 60% | 48% |
| Incomplete clusters | 0 | 32 | 0 | 32 |
| Over complete clusters | 0 | 102 | 0 | 102 |

**Table 6.3**: results of a more advanced single document NE co-reference resolution method

The goal of this 'Advanced' method was to raise the recall while keeping the high precision. This goal was partially met; the recall was raised by almost 0.2 points but the precision was lowered by 0.07. The percentage of clusters that are fully correct has almost doubled indicating that the new method solved a lot of the previous problems. Unfortunately the precision is lower (a common effect when optimizing recall).

### 6.3.3 Discussion of the results

The main drawback of this method is the lower precision score. This can be explained by looking at the new comparison method. If two names share one single word (no matter how long the names are) they are clustered together. After looking at the clustered names there are some mistakes that caused this problem:

- Names that share a common first or last name (Jansen, Jan, Frits, Wouter, Fries etc).
- Names that share initials (often only one letter)
- Titles of persons (de heer, mevrouw, minister etc.)

Concluding based on only one word out of a name whether two names are co-referential can introduce wrong names into a cluster. This indicates that the other words are also needed to indicate if two names are co-referential or not.

The recall was raised but still is not perfect. The main reason is yet again the strict comparison method used to compare parts of the names. In some cases names that are co-referential have slightly different surface forms resulting in a mismatch. There are two different problems that cause this problem:

- Spelling variations or spelling errors (like Fortuijn, Fortuyn)
- Mistakes made by the NNER that change the surface form of the names (see chapter 3).

These problems need to be handled in order to get a higher recall.

## 6.4 JW-method

The main reason for the 'failure' of the advanced method is that it bases its classification solely on parts of the name and not on the entire name. A second problem is spelling variations in the names. There are other string comparison methods that give an indication of the similarity of two words instead of a boolean match. The following method uses one of these more elaborate methods. The hope was that such a method would raise the precision with a minimal impact on the recall score.

### 6.4.1 Selection and implementation of JW-method

Because string distance methods are used a lot in a different number of fields (information retrieval, text mining, natural language processing etc) they have been studied and evaluated extensively. Examples are recent work from Cohen et al. evaluating a great number of methods used for string comparison (Cohen, Ravikumar and Fiendberg; 2003) and work from Branting who evaluated name-matching methods (Branting; 2003). The best performing method described by Cohen et al. is a TFIDF-JaroWinkler method. However, this method seems to work best on strings containing a large number of words rather than on typical name strings with only few words. The methods proposed by Branting for name comparison seems to work very well for names containing few words and his work gives more insight in the mechanisms used than Cohen does.
For this method a simpler version of the best method described by Branting is implemented consisting of two steps:

- Normalization: removal of punctuation, space-normalization, abbreviation replacement, capitalization and stop-word removal.
- Similarity assessment: similarity measurement between two names.

The first step is already done in the advanced method and will be used in the exact same way for this method. The second step is similarity assessment of two names and is not really described by Branting. Two commonly used string similarity methods are Jaro and JaroWinkler. The Jaro similarity (Jaro; 1989) is based on the number of matching characters and gives a score between zero and one (the higher the score the more two strings are the same).

Both methods were implemented and tested on 100 name pairs extracted from the corpus. Half of these name pairs were meant to be similar, the other half should not be found similar. The Jaro-Winkler method slightly outperformed the Jaro method and hence it was used for string comparison.

Similarity assessment is done using Jaro-Winkler measurement on each pair of words contained in two names. The maximum score for each word in the name that has the smallest number of words is remembered and in the end combined into a single similarity score.

$$Final\ score = \frac{\sum s}{n}$$

$$s = set\ of\ \max.scores\ \ for\ each\ word$$

$$n = number\ of\ \ words$$

As with the baseline method names need to be compared to clusters. The comparison-method used for the baseline turned out to be best for this method as well tested on the same data as described in section 6.3.1.

The handling of abbreviations is not really described by Branting other than the fact that they are normalized. In this work abbreviations are handled in the same way as normal name comparisons. Every letter of the abbreviation is considered to be a separate word and is compared with the first character of every word within the second name. This also works for abbreviations within a name, like initial letters of first names of a person. For example the names "Balkenende (J.P.)" and "Jan Peter Balkenende" will be matched with a high score since the last names match perfectly and the abbreviation "JP" matches the first characters of "Jan Peter".

### 6.4.2 Evaluation JW method

The evaluation of the JW-method was done in the exact manner as the evaluation of the baseline method. Also, the exact same files were used.

**Results**

The following table (6.4) shows the average precision, recall and F-measure scores for the 70 evaluated documents (for both ground truth and NNER annotated sets). The results of the JW- method are presented in the last two (shaded) columns called JHAN and JNAN. The explanation of these results can be found in the discussion section (6.4.3)

| Evaluation metric | BHAN | BNAN | AHAN | ANAN | JHAN | JNAN |
|---|---|---|---|---|---|---|
| Precision | 0.99 | 0.99 | 0.95 | 0.91 | 0.93 | 0.94 |
| Recall | 0.73 | 0.65 | 0.88 | 0.82 | 0.89 | 0.81 |
| F-measure | 0.84 | 0.78 | 0.91 | 0.86 | 0.91 | 0.87 |
| Fully correct clusters | 37% | 26% | 60% | 48% | 57% | 48% |
| Incomplete clusters | 0 | 32 | 0 | 32 | 0 | 32 |
| Over complete clusters | 0 | 102 | 0 | 102 | 0 | 102 |

**Table 6.4**: results of the JW-advanced single document NE co-reference resolution method

The most important result is the precision growth of 0.3 points on the NNER annotated names while the recall got lowered by 0.1 points (with respect to the advanced method). This was the main goal of this method. Strangely enough this method had the opposite effect on the 'human annotated data'; the precision got lower while the recall got higher.

### 6.4.3 Discussion of the results

The two most important improvements: combining the scores from all the word-comparison and usage of a different string comparison method improved the precision on the NNER annotated names. The main reason for this effect is that grouping two names was not based on the best match of words within the names but on weighted sum of similarity scores. Using this method names that were erroneously grouped before because they partially matched are no longer matched together. Fortunately this change did not affect the recall.

It is very strange that the opposite happened using the human annotated names. Apparently the less strict JaroWinkler method erroneously clustered names together. It is not likely that this solely happened on human annotated data. Instead it is more likely that it also happened using NNER annotated data but that it fixed more precision problems due to NNER mistakes than that it caused mistakes. So the JW-Advanded method did more right than wrong on the NNER-data while it did more wrong than right on the human-data. However this argumentation is just an 'educated guess' and is not proven.

The problems stated in the baseline discussion that were not addressed by JaroWinkler method still exist. These are mostly hard cases of exact the same names that refer to different things (semantic ambiguous names), or different names that refer to the same entity. Some names are non rigid designators (like titles) and these are very hard to match with other names.

Looking at these problems it is not likely that string based methods like JW-advance method will be able to do better. More information is needed to cope with semantic ambiguous names or with homonymic names. Since it was the aim to find a method solely based on 'cheap' string comparison methods there is not reason to try to improve this method more.

## *6.5 Using name type information*

Since the corpus also contains the types of the names it is possible to use this information for the co-reference resolution. This is of course not realistic with regard to the available system since it can not do name-classification. The types in the corpus were hand crafted and hence have a very high. Nevertheless it was very interesting to see the impact of the availability of such information.

One very simple heuristic was added to the methods described above: *Only add a name to a cluster if it has the same type as the items in the cluster.*

So every name within a cluster must have the same type. This is very logical since two names (even if they are exactly the same name) can not co-refer to the same thing if they have of different types.

The presumed effect of this method should be a higher precision and recall scores. The intuition behind this assumption is the fact that two similar names with different types are no longer clustered together. In this way highly ambiguous names (like nation names) will be resolved with a higher accuracy.

### 6.5.1 Evaluation of this type-enriched-method

The evaluation was done in the exact same way as the above described evaluations. Note that this evaluation could only be done on data with type information, hence only 'human annotated names' could be used. Table 6.5 has two columns for each method, one with a +T and one without. The columns with the +T have the scores of the methods with the type information (B = Baseline, A = Advanced and JW = JaroWinkler).

| Evaluation metric | B | B+T | A | A+T | JW | JW+T |
|---|---|---|---|---|---|---|
| Precision | 0.99 | 0.99 | 0.95 | 0.97 | 0.93 | 0.97 |
| Recall | 0.73 | 0.73 | 0.88 | 0.89 | 0.89 | 0.90 |
| F-measure | 0.84 | 0.84 | 0.91 | 0.93 | 0.91 | 0.93 |
| Fully correct clusters | 37% | 37% | 60% | 65% | 57% | 66% |
| Incomplete clusters | 0 | 0 | 0 | 0 | 0 | 0 |
| Over complete clusters | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 6.5**: results of the different methods with and without NE-type information

These results are very interesting because the baseline method did not improve while the JW-method improved quite a bit. The JW-method performed better with the type information than without. Especially the precision score is better and the percentage of fully correct chains grew with 9%. Using type information had less effect on the Advanced method.

### 6.5.2 Discussion of the results

The results show an improvement of both the advanced- and JW-method but not for the baseline method. In fact the precision of the baseline method improved a little bit, but not enough to become 1. This means that the usage of type information did 'disambiguate' a small portion of the names that are exactly the same but apparently this does not occur a lot. The reason that the recall did not change is very simple; names that were not clustered together because of a difference in surface form are still not clustered together.

The results show an overall improvement of the Advanced and JW- methods. However it helped the JW method more than it helped Advanced method. In both cases type information was used to disambiguate names with a high similarity. Names that do not match a group based on their type could now be added to another group which causes the recall to become slightly higher.
The precision scores of these methods come really close to the precision the human annotators achieved amongst each other. The recall scores are still not as good but this is not really strange. Names that have very low similarity will not be grouped together even if they have the same type. So using type knowledge does not help entities with totally different names, at least not in the simple way they were used now.

These results are a strong indication of the usefulness of named entity type information in addition to string matching algorithms. In this evaluation this information was used as a constraint. However, type information can also be used to do a more specific string distance measure. For example specific rules can be used to compare two names; if it is known that a name belongs to persons the internal structure of the names can be used for a more proper similarity metric.

## *6.6 Complexity of the solutions*

The complexity and scalability of the different solutions was not used as selection criterion and got minimal attention. This section will briefly address the complexity of the methods and assess the influence of the NNER on the total processing time.

From a theoretical point of view the complexity is linear in the amount of documents since the all the documents are processed individually. If the number of documents doubles the time needed to process them also doubles. This means that the processing time depends on the complexity of the processing done on each document. Unfortunately it is very hard to do a purely theoretic complexity assessment of the single-document NE co-reference resolution methods since the NNER also takes time. If the complexity of the different resolution methods is neglectable with respect to the NNER their complexity will be less important.

The processing time of the three different methods was recorded with and without the NNER (GATE reports the total time needed to process a set of documents). Each individual measurement was done three times under normal circumstances to minimize the effect of processes running on the computer. The results are shown in table 9.1, the columns show the number of documents processed. Columns with a '+n' show the processing time including the NNER. All the values given are seconds.

| | 70+n | 70 | 140+n | 140 | 215+n | 215 |
|---|---|---|---|---|---|---|
| **Baseline** | 65 | 7 | 139 | 16 | 212 | 25 |
| **Advanced** | 82 | 19 | 162 | 51 | 251 | 81 |
| **JW-method** | 93 | 31 | 185 | 79 | 296 | 125 |

**Table 9.1**: processing times of the three single document NE co-reference resolution systems for three different sets of documents (on a 800Mhz, 512MB machine)

Using these results a number of important things can be calculated:

- The time it takes for the NNER to process a document is 0.8 seconds on average
- The time it takes for the baseline to process a document is roughly 0.12 seconds on average
- The time it takes for the Advanced method to process a document roughly 0.35 seconds on average.
- The time it takes for the JW-method to process the document is roughly 0.55 seconds on average

These results show that the NNER takes up most time which lowers the influence of the different resolution methods on the processing time. The JW-method on its own is 5 times slower than the baseline method, but when used in combination with the NNER this is only 1.33 times slower.

The Advanced method is slower than the Baseline but faster than the JW-method. This can be explained by looking at the way those methods compare the names. The comparison strategy for all the methods is the same. Basically all names are compared with all the previous encountered names within the document. This takes n*(n-1)/2 comparisons which means it is in the order of $n^2$ ($O(n^2)$) where n is the number of names in a document. This means that only the comparison methods can account for the difference in processing time.

The baseline method uses the java string equals method which simply compares the $i^{th}$ character in $name_1$ with the $i^{th}$ character in $name_2$. However, this method is optimized to stop when two characters are different or if the two strings have a different length in the first place (which happens most of the time). This results in an average complexity of O(1) with a worst case complexity of O(n) where n is the number of characters.
The advanced method also uses this equals method but not on the entire names but on all combinations of words within two names. So instead of using the equals method only once for each name pair the equals method is called k*l times where k and l are the number of words in respectively $name_1$ and $name_2$. Looking at the results l*k is 3 (0.35/0.12) on average which is logical because names typically consist of 1, 2 or 3 words.
The JW-method is on average 5 times slower than the baseline. This is hard to explain because the method does a lot of operations. The most expensive one matches each character in $word_1$ with a range of characters in $word_2$. This range depends on the length of the two words. Apparently this range is not that big on average, else the processing time would have been a lot larger.

## *6.7 Conclusion*

Looking at the precision it can be concluded that the hypothesis stated at the beginning of this chapter is valid. Names within the same document that have very similar surface forms are very likely to be co-referential. The problem is how to do the string matching in such a way that most co-referring names are found without lowering the precision.

The study described in this chapter used a lot of different ways to compare names:

- Compare the full names, compare individual words in the names and use normalization on the names.
- use different methods to compare the different parts of the names (very strict or less strict)
- how to define the similarity between a name and a group of names

The best performing method averaged the individual similarity scores of all the names in a cluster, used simple normalization and compared words using the JaroWinkler method. This JW-method has a slightly lower precision score but a lot higher recall score and percentage of fully correct clusters than the baseline method currently used by Novalink.

Even though this combination performs very good there are still two types of problems that can not be solved in this way:

1. Two different names that are co-referential:
   - Non-rigid designators like 'names' with titles (mayor of Amsterdam)
   - Entities that have multiple names like nicknames, separate usage of first- and last name ('J.P.', 'Balkenende' and 'Harry Potter')
2. Two similar names that are not co-referential:
   - Very common names like 'Jansen', 'Vries' etc. (very rare in single documents)
   - Semantic ambiguous names, often location names ('Nederland' referring to the Dutch government, 'Nederland' referring to the population, 'Nederland' referring to a sports team).

Looking at these problems it is unlikely that a method strictly based on string comparisons will improve the current results. The mistakes made are mostly 'hard' cases where information other than the surface form of the string is needed to identify and solve them. The small study done on methods using such knowledge, in the form of the name-type, already showed an improvement over the string-comparison methods.

Other possibilities are:

- Use comparison of (local) context in addition to the comparison of surface forms (nouns in the sentence, verbs associated with the name etc). Information like this could help solving problem(s) 1.
- Use semantic information in the form of dictionaries, synonym lists to find common alternative names for a given name ('Oranje' and 'Nederlands elftal'). This kind of information could help to disambiguate similar names (problem 2).

The complexity assessment of the different methods with and without the NNER showed that the JW-method on itself is 5 times slower than the baseline method. However, the NNER uses most of the processing time diminishing the effect of slower NE co-reference resolution methods. The JW-method will increase the time needed to process a set of documents by 30%. It depends on the effect on the text mining if this is acceptable or not. There are probably ways to make the JW method faster (estimate if two names should be compared using the JW-method in the first place).

# 7 Cross document named entity co-reference resolution

## 7.1 Introduction

The co-reference resolution method as described in the previous chapter only attempts to solve name co-references in single documents. This is only part of the solution. A lot of co-referential names occur in different articles and these still need to be found. This chapter describes the study done and methods used to solve these 'cross document named entity co-references'. The starting point of these solutions, except for the baseline method, is the set of co-reference clusters as created by the single document JW-method. The goal of this study is to answer the research questions regarding the usefulness of features that can be used for cross-document named entity co-reference resolution.
When matching the clusters of names it is probably not good enough to only use the surface forms of the strings again. The assumption that similar looking names co-refer, as used in the previous chapter, is no longer a safe assumption since the names originate from different articles. Other information is needed to be able to compare the clusters.
The work done described in this chapter focuses on finding a good similarity metric and does not address the scalability problem. All the clusters of names are compared with each other resulting in a complexity in the order of $n^2$ (which is hardly scalable). The reason for this approach is that it enables methods to find all the co-referential names, in contrast with methods that do some preliminary selection.

This chapter describes the work done to define and asses the usefulness of different pieces of information (features from the articles and names) for cross-document co-reference resolution. The features used for this study originate from literature or by simply using common sense. Statistical analysis was used to predict the usefulness of the different features. Machine learning techniques were used to train different classifiers on (sub) sets of the features. The evaluation of these models gave a second indication of usefulness of the features.

The single document named entity co-reference resolution methods were evaluated on 'human annotated data' and on 'NNER annotated data' (as proposed by Mitkov). However doing the same for cross document named entity co-reference resolution would be extremely time consuming since the training of models would need to be done twice. So only NNER annotated data was used for this part of the research giving an indication of the real performance of the method.

The reader should bear in mind that during the study described in this chapter compared *clusters of names* and not individual names (except for the baseline method described in the next section). These clusters of names can vary in number and variety of names. Each cluster originates from only one document.

## 7.2 Baseline method

A baseline method was created to be able to asses the usefulness of other methods. The same baseline method as used for single-document co-reference resolution can be used for the cross-document study. The same basic rule was used:

*Two names are co-referential if they are exactly the same.*

Note that this baseline is *not* based on the clusters as found by the JW-method but simply groups all the names in all the documents together if they are exactly the same (case sensitive).

### 7.2.1 Results

The baseline method was evaluated on 215 documents (no 'tuning' was done so the majority of the documents was used). The results are shown in the table 7.3 below. As described in chapter 4 clusters containing only one name are omitted from the evaluation.

| **Evaluation metric** | **NNER annotated names** |
|---|---|
| Precision | 0.86 |
| Recall | 0.67 |
| F-measure | 0.75 |
| #clusters 100% correct | 21% |
| #clusters 100% precision | 70% |
| #clusters 100% recall | 30% |

**Table 7.3**: results of the baseline cross-document NE co-reference resolution method

These results are very similar to the baseline results of the single document method; a 'high' precision and lower recall scores. In addition to the number of clusters that are 100% perfect also the percentage of clusters with perfect precision and recall scores are given. This is done to get a better understanding of the quality of the system. Now it can be assessed if low scores are a result of few mistakes in big clusters or if they are a result of a lot of small mistakes in a large number of clusters.

### 7.2.2 Discussion of the baseline results

The precision and recall scores in combination with the percentages of clusters that are perfect indicates that mistakes were made in most clusters but that these mistakes are not 'big' with respect to these clusters. If the mistakes made by the baseline method were bigger the precision and recall scores would have been a lot lower. Since 70% of the clusters have perfect precision the main problem is the low recall. Apparently a lot of clusters miss names.

The fact that the precision score in the cross-document step is lower than the precision score in the single document step is an important observation. This indicates that the rule used is 'less valid' in case of names occurring in different documents than names that occur in the same document.

## *7.3 Using additional information*

### 7.3.1 Introduction

Using only name similarity is no longer enough and extra information, in the form of features, is needed. One of the research questions concerned the selection, usability and complexity of features that can be used for named entity co-reference resolution. In the case of this graduation research features are similarity or distance scores of some aspects of two clusters. These scores can then be used to determine whether the two clusters are co-referential or not.

The set of features proposed in this section were mostly inspired by literature and translated to this domain (if needed). A dataset was build using these features and a preliminary statistical analysis of this data was done to see what the data looks like.

### 7.3.2 Selection and similarity measurement of features

First of all the possible usable features must be selected. Some of the features were selected using common sense thinking while others were found in literature. One piece of information commonly proposed in literature is the usage of 'context' for textual analysis ((Baldwin and Bagga; 1998a), (Niu, Li and Srihari; 2004) and (Lee, On, Kang and Park; 2005)). What this context is largely depends on the kind of documents used. For example Lee et al. use co-authors, conference names and title names as features to disambiguate authors in digital libraries. Peng et al. (Peng, He and Mao; 2006) use local- and global context to disambiguate location names. I think that some of these 'features' can also be used to solve cross-document named entity co-reference resolution.

Machine learning methods need numeric features for training and validation. So the similarity of the different features needed to be calculated and put in a useful format. An important aspect of these calculations is their complexity. A lot of these calculations need to be done in order to compare all the clusters, so fast calculations are preferred.

**1 - Names similarity**

The similarity between names has been the most used feature in this research and will also be used as a possible feature for the cross document resolution. Again, the idea behind this is that names that look similar have a high chance to co-refer. However this idea should be used with caution since it might not be valid for all names in different documents. Other features are needed to help out in this case.
Since it is no longer about single names but about clusters of names the strategy used in the single-document step can not be used. Names in a cluster can be represented as a vector and the similarity of names in two clusters can defined by the similarity of two name-vectors. This similarity can be calculated in a very straight forward way. The number of words that occur in both sets is divided by the size of the smallest set. In this way the similarity score is always a number between 0 and 1 (the higher the score higher the similarity). For example the similarity of the following sets of elements {A, B, C} and {A, B, D, E} is 2/3 because 2 out of the 3 elements are the same.

**2 - Global context**

As proposed in literature the global context of the documents is a very valuable source of information which should be used. This context is typically a set of words that are representative for the text. However, the extraction of these words is hard. TNO-ICT developed a tool that does this job; it classifies texts with labels from a news thesaurus. These labels do not necessary have to occur within the text. This has the advantage that texts that are roughly about the same things get the same labels. For example crime and law-suits can have labels like: police, criminal, extortion, judge etc. So for a cluster of names the labels extracted from the documents these names occur in can be used as 'global context'. The top 30 labels were used. These labels can also be represented as vectors and the same similarity metric as used for feature 1 was used.

**3 - Name co-occurrences**
Another type of context can be the names that often co-occur with a cluster. For example a cluster of names concerning the NE 'G.W. Bush' can have a set with frequently co-occurring names like {Washington, United States, Iraq}. The intuition behind this feature is that two names which have 'high' similarity between the sets of names they frequently co-occur with might be co-referential. If for example another cluster with 'Bush' has a set of names like {Amsterdam, Police, …} this is an indication that the two Bush clusters are not co-referential.
Novalink is largely based on this idea. It is not really the case that co-reference of two names can be based solely on co-occurrence of other names. But it might be a usable feature which gives an indication for co-reference or not. The selection of this feature is also inspired by the fact that names are already present in the system so no extra parsing needs to be done. Since the initial clusters of names are extracted from only one document all the names within that document were used (which can be a lot or just a few depending on the document size). These names can also be represented as vectors and the same similarity metric as used for feature 1 was used.

**4 - Difference in publish-dates**
Every cluster has a date when the name occurred first in the news and a date when it occurred last in the news. This time information can also be used as a piece of information to match clusters. The intuition here is that 'entities' tend to be in the news for a period of time and not continuously. This information can be used for the disambiguation of two similar names. If for example two clusters with common names are compared, and one cluster occurred in the news one year before the other cluster this is an indication that they are not co-referential. Unfortunately this observation does not always hold; important people and locations (like prime ministers) tend to be in the news 'continuously'.
The 'temporal distance' between two clusters can very simply be defined by the number of days that lies between the publish dates. For two clusters; names in cluster one were published at 01-04-2000 and names in cluster two published at 11-04-2000 the difference is 10 (days). If the dates of two clusters overlap the distance is 0. Features 1 to 3 lay between 0 and 1 while feature 4 can have any positive number. To make feature 4 more usable for machine learning techniques it was normalized to a number between 0 and 1 by dividing the original number by the maximum (which was 4000 days).

Other possible features were article source and author. However, these features are very general in the context of news articles and hence were not used (there were no differences observed during the annotation work). Different sources tell the same news-stories and names are used in the same (formal) way.

### 7.3.3 Statistics on the features

The dataset was analyzed to see what the data looks like and to see how distinctive the features are for the positive and negative samples. A good visualization are the box plots shown in figure 7.3 and 7.4 (made with Matlab) based on 5000 positive and 5000 negative samples. Figure 7.3 shows the distribution of the values for the positive samples and figure 7.4 shows the distribution for the negative samples. A lot of difference between the distributions is good because that indicates that the features can be used to separate the positive from negative samples.

The biggest difference is the name similarity which is significantly higher for the positive samples. The positive name similarity mean of 0.6 is a bit low but can be explained by the strict comparison method used. Clusters typically only contain very few words with different spelling (permutations) and anomalies introduced by the NNER. This makes it less likely that all the names in two clusters match exactly.

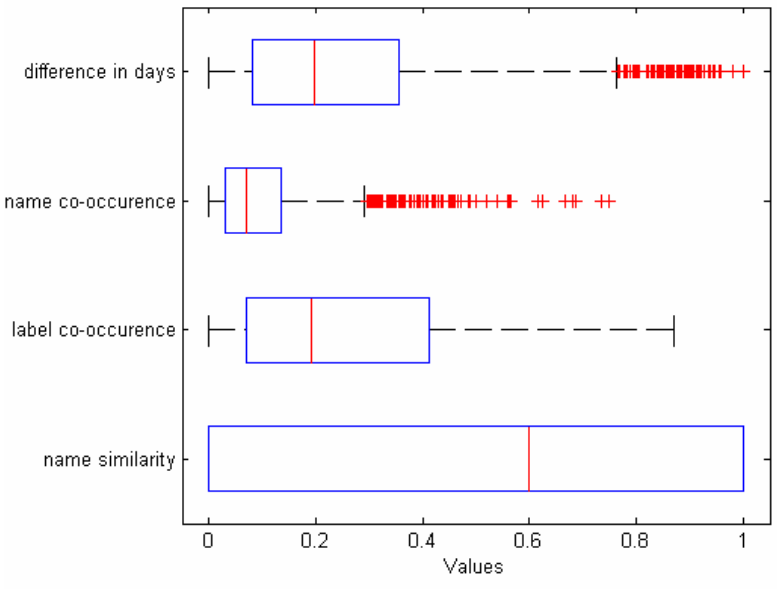



**Figure 7.3**: distribution of the data from positive samples

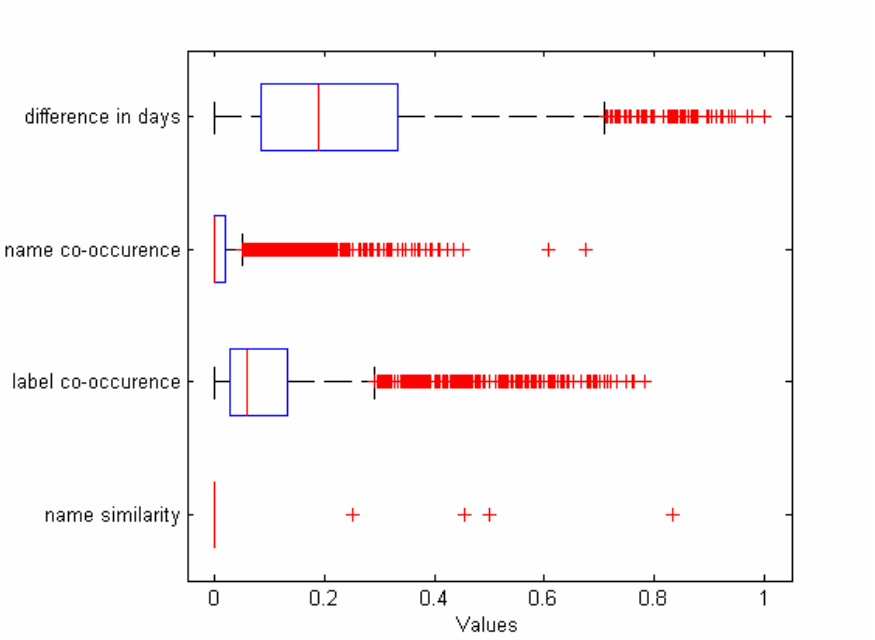



**Figure 7.4**: distribution of the data from negative samples

The medians of 'name co-occurrence' and 'label co-occurrence' are lower for the negative samples. However, these features are certainly not as distinctive as the name similarity. The 'difference in days' distributions appear to be largely the same and hence will probably have very little distinctive 'power'. These observations do not say a lot about the strength of the different combination of features because the plots are independent of each other. Other study was needed to assess the combinations of features, see chapter 7.5.

The box plots show a great number of outliers, especially for the negative samples. The samples where these outliers belong to can be exceptions which are harder to learn (or classify correctly).

It is also important to know the true distribution of positive and negative samples. This gives insight into the task that needs to be learned and that distribution should also be used for evaluation of classification models. Since most names do not co-refer the percentage of negative samples should be really big. The ground truth data was used to calculate this distribution. In reality 99.9% of all samples is negative and only 0.1% is positive. This means that from all comparisons done only 0.1% is done between clusters that co-refer (under the assumption that all possible cluster combinations are compared).

## *7.4 Using Machine Learning to Train models*

### 7.4.1 Introduction

It is very hard to find the best settings for the features by hand. A rough estimation can be done to assess the usefulness of the individual features but it is to time consuming to do this by hand. Hence, machine learning techniques were needed to do this.

The last part consists of training and evaluating several models using machine learning. Different subsets of the selected features were used and the evaluation of these models also gave an indication of the contribution of the individual features. There was one validation set consisting of 100.000 negative and 100 positive samples used for all the evaluations. The reason is that there were not enough negative samples to make more validation sets.

### 7.4.2 Similarity estimation

A simple and intuitive way to compare two clusters is to find a function that approximates the similarity of the clusters using the features as input parameters. A threshold can then be used to determine if the similarity is high enough and hence 'classify' the two compared clusters to be co-referential. The simplest function presumes a linear relation from the parameters to the similarity score. In such a function the outcome is the sum of the product of weight – feature pairs (possibly added with a constant). In this case a linear function will have the form:

$$Similarity = w_1 * f1 + w_1 * f2 + w_1 * f3 + w_1 * f4\ (+b)$$

The assumption is that some features are more important for the similarity assessment than other so their weights must be higher. Weights can also be negative (w4) to give a penalty for high feature values. The trick is to find the best set of weights with an additional threshold which separates the negative from the positive samples. The simplest method is to let a computer try out a great number of weight-sets and thresholds in a controlled way.
This is not really a machine learning method other than the fact that the optimal weights were found by the computer. This method was tried to see the performance of a simple linear model and get an impression of the importance of the different features.

To find the best weights they were systematically changed. For each set of weights the best threshold was defined using a training set (9000 positive and 1000 negative) and evaluated on the evaluation set (100.000 negative and 100 positive). The weights ranged from 0 to 5, resulting in total of 1295 different permutations ({0, 0, 0, 1} to {5, 5, 5, 5}). Even though this is not a guarantee to find the optimal set of weights it still gives a good estimation of the optimal ratio between the weights. The optimal results are presented in table 7.5 (below).

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.54 |
| **F-measure** | 0.50 |

**Table 7.5**: best results of similarity estimation

There are several weight sets with this same result, all these sets have high features 1 and 4 and zero for features 2 and 3. More precisely, this result is mostly met (but not always) if features 1 and 4 are within 1 point of each other ({2, 0, 0, 3} but also {3, 0, 0, 3}). Slightly lower scores (F-measure between 0.47 and 0.49) are achieved with high features 1 and 4 and low features 2 and 3 ({5, 1, 0, 2}). This indicates a higher importance of feature 1 and feature 4. More general, feature 1 needs to be involved (strongly) in the function to get reasonable results. Strangely enough feature 4 on itself is not a good feature, but using feature 4 in addition to 1 accomplishes the best results.

### 7.4.3 SVM classification

A more advanced classification method is a Support Vector Machine (SVM). A SVM maps training data to N-dimensional space and tries to find a set of n-1-dimensional hyperplanes that separates the different classes in the data. There can be a lot of different sets of hyperplanes that do this, SVM's try to find the best set that separate the different classes with maximum margin. The original algorithm was a linear classifier because a dot product was used to map the features into the n-dimensional space (same as the method described in 7.5.2). However, current SVM's can use non-linear kernel functions to map the data to higher dimensions instead of the dot product.

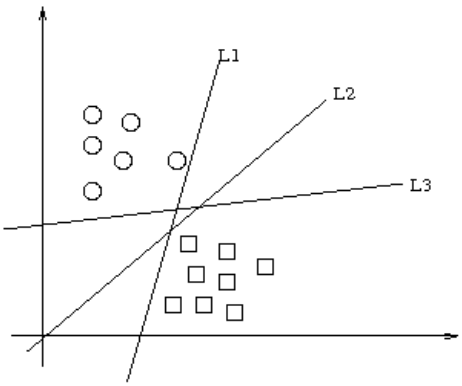



**Figure 7.6**: three hyperplanes that separate the data, but only L3 achieves maximum separation (source: wikipedia.org)

The package used for training and evaluation of SVM's in this research was LIBSVM (Chang and Lin; 2001). LIBSVM has a good manual, is broadly supported and is available in multiple programming languages (in source and binaries).

The methodology used to train SVM's largely follows the methodology proposed by the creators of LIBSVM (Chang and Lin; 2007). In an iterative process models were improved on F-measure of correct classification of co-referential clusters (and not accuracy). The reason is that it is important for the text mining process to keep the clusters as correct as possible. Clusters with members that do not really belong in the cluster can:

- Cause more faulty names to be included in the cluster
- Cause erroneous text mining results because they are based on impure data.

The first SVM was trained on all the features using a dataset with 9000 negative and 1000 positive samples. A RBF kernel was used with standard gamma and Cost settings (g = 1/k, C = 1). The training and evaluation was done three times on randomly extracted data sets. The average results are shown in table 7.7.

| | |
|---|---|
| **Accuracy** | 99.88% |
| **Precision** | 0.46 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

**Table 7.7**: the optimal results for SVM classification using all the parameters

These results show really high accuracy but relatively low precision and recall scores. These scores indicate that 44 of the positive samples are classified negatively and 64 negative samples are classified as positive. Looking at the total number of samples this is only 0.12%, but the number of misclassified samples is enough to cause the low precision and recall scores. This result is very similar to the result of the 'linear similarity estimation' method. This indicates both methods identify the same 'hyperplane' (most likely heavily based on feature 1).

Since normal SVM's optimize their model using the accuracy it is hard to improve on this result. Still some options are available to improve this SVM result:

- Optimize gamma and cost parameters for SVM kernel
- Use one-class SVM model
- Optimize SVM on F-measure and not on accuracy

Unfortunately none of these methods improved on the basic method described above. The optimizations tried are shortly described in appendix C.

SVM's were also trained on different sets of features to assess the strength of these combinations. These different sets are all the permutations containing feature 1. All the SVM's were trained on 9000 negative and 1000 positive samples and evaluated on the exact same set as used before. In the trainings set the positive samples were over sampled as suggested by (Hulse, Khoshgoftaar and Napolitano; 2007). They studied different solutions to cope with imbalanced data for different machine learning methods. The best method for SVM classifiers was '1000 time random over sampling'. In this study 100 times over sampling performed the same as 1000 times over sampling and hence the above mentioned trainings set was used. The results are shown in tables 7.8.
A small study of SVM's trained on data without feature 1 resulted in majority classifiers where all samples are classified as negative.

Results using only feature 1

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

Results using features 1 and 2

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

Results using features 1 and 3

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

Results using features 1 and 4

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

Results using only feature 1, 2, 3

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

Results using features 1, 2, 4

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.46 |
| **Recall** | 0.55 |
| **F-measure** | 0.50 |

Results using features 1, 3, 4

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.47 |
| **Recall** | 0.56 |
| **F-measure** | 0.51 |

**Tables 7.8**: SVM classification results using different sets of features

All these results show that features 2, 3 and 4 do not contribute a lot (or nothing at all) to the final classifier. All the SVM's perform exactly the same or slightly worse than the SVM trained using only feature 1. This indicates that features 2, 3 and 4 are too 'weak' compared to feature 1 to train a good classifier.

### 7.4.4 K-nearest neighbour classification

The two methods described above basically try to learn rules used to classify data. Another method is to compare unseen data with data with known classes. The unseen data is classified the same as the data it matches best with. The K-nearest neighbours (KNN) algorithm simply does a majority vote among the K nearest neighbours of the new data (where K is a predefined numeric parameter). The algorithm learns how to compute the distance between data points to get the best classification accuracy. This learning is done by determining weights for each feature.

In case of an even K ties are possible and are mostly resolved randomly (or other information like more neighbours or class distribution can be used). If for example a new data sample has 7 neighbours; 4 positive and 3 negative the new sample will be classified as positive (see also figure 7.9).

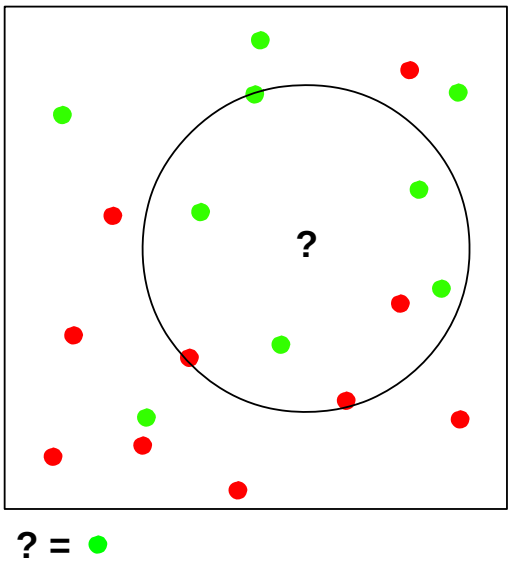



**Figure 7.9**: Example 7-nearest neighbour

The KNN classification package Timbl (Daelemans, W. and Van den Bosch, A.; (2005)) was used to train a good model. With Timbl a large number of classification algorithms, optimizations and parameters can be set.
The distribution of the training data is very important because the final classification is based on the number of neighbours. Especially in this case with an extreme unbalanced dataset it is hard to predict the right distribution of training data. Zhang et al. (Zhang and Maini; 2003) investigated the best distribution of (extreme) unbalanced sets and concluded that under-sampling of the majority class by using only 5% worked best. A small preliminary study using this conclusion showed the best results using 10% under-sampling (10000 negative, 100 positive).

Timbl offers a large set of different learning algorithms with different weighting schemes. A large set of different configurations were tried:

- K {1,3,5,7}
- Weight metrics: {no weighting, gain ratio, Chi-squared}

The optimal settings found are K of 3 (although 5 and 7 had the same performance) and Gain Ratio weight metric. The classification of the 100100 evaluation samples took a lot of time when the standard KNN algorithm was used (up to 6 hours). Instead the TRIBL2 classification method was used. TRIBL2 is a hybrid between the slow KNN algorithm and fast IGTREE method (which is a fast tree structure).

This setting (k=3, gain ratio weighting, numeric features, TRIBL2 classification) gave the results shown in table 7.10:

| | |
|---|---|
| **Accuracy** | 99.80% |
| **Precision** | 0.30 |
| **Recall** | 0.73 |
| **F-measure** | 0.42 |

**Table 7.10**: the optimum KNN classification results (average of three runs)

| | |
|---|---|
| **Feature 1** | 0.5507 |
| **Feature 2** | 0.0020 |
| **Feature 3** | 0.0059 |
| **Feature 4** | 0.0005 |

**Table 7.11**: information gain values found by the KNN algorithm

These results again show very high accuracy but lower precision and recall scores. This method has higher recall and lower precision scores than the previous machine learning methods. This indicates that the KNN algorithm has the tendency to classify more samples as positive.

Timbl also shows the Information Gain which is a measurement how much information each feature contributes to the knowledge of the correct class, see table 7.11. So these values give the contribution of each feature for the classification task.

These scores show the importance of feature 1, same as the previous methods. Features 2, 3 and 4 seem rather insignificant with respect to feature 1 and hardly contribute to the classification. It should be noted that these values are computed for each feature independently. This can explain the difference in weights with the 'similarity estimation' which indicates feature 4 to also be important (in combination with feature 1). To assess this, the KNN algorithm was also used to classify samples using all feature combinations that have at least feature 1, see tables 8.12.

Results using only feature 1

| | |
|---|---|
| **Accuracy** | 99.90% |
| **Precision** | 0.50 |
| **Recall** | 0.70 |
| **F-measure** | 0.58 |

Results using features 1 and 2

| | |
|---|---|
| **Accuracy** | 99.80% |
| **Precision** | 0.30 |
| **Recall** | 0.73 |
| **F-measure** | 0.42 |

Results using features 1 and 3

| | |
|---|---|
| **Accuracy** | 99.86% |
| **Precision** | 0.38 |
| **Recall** | 0.68 |
| **F-measure** | 0.49 |

Results using features 1 and 4

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.49 |
| **Recall** | 0.74 |
| **F-measure** | 0.59 |

Results using only feature 1, 2, 3

| | |
|---|---|
| **Accuracy** | 99.89% |
| **Precision** | 0.48 |
| **Recall** | 0.68 |
| **F-measure** | 0.56 |

Results using features 1, 2, 4

| | |
|---|---|
| **Accuracy** | 99.87% |
| **Precision** | 0.41 |
| **Recall** | 0.71 |
| **F-measure** | 0.52 |

Results using features 1, 3, 4

| | |
|---|---|
| **Accuracy** | 99.82% |
| **Precision** | 0.31 |
| **Recall** | 0.68 |
| **F-measure** | 0.42 |

**Tables 7.12**: KNN classification results using different sets of features

As with the 'similarity estimation' feature 1 and the combination of 1 and 4 give the best result. Feature 4 seems to be a valuable addition to feature 1. All the other feature combinations score either the same or better than the full combination. It seems that certain feature combinations are bad for the classification performance. While the combination 1 and 4 performs best the introduction of feature 3 lowers the F-measure by 0.17 points! more?

### 7.4.5 Discussion of the machine learning results

Even though the classification accuracy of all of the models learned using machine learning techniques are very high they are inappropriate for accurate classification of clusters that should be combined. Both the precision and recall do not outperform the baseline, especially the precision is very low.
There are three causes:

1. The highly imbalanced data
2. Insufficient strength of the features
3. Small number of features

As explained in the dataset section (7.3.3) the data is highly imbalanced (10000 negative sample for every positive one). It is a known problem that standard classification methods have poor performance on this kind of data. The reason is that most machine learning techniques "maximize the overall accuracy leading to 'trivial' classifiers that tend to ignore the minority class" (Zang and Maini; 2003). The low SVM recall for the positive samples also shows this (the negative samples are a lot better classified).

Another very important reason is the insufficient 'distinctiveness' of some of the four features. All the methods show the importance of feature 1, which is by far the most distinctive feature. Unfortunately the other features are not strong enough to classify the exceptions correctly (those that should be classified differently than their name similarity suggests).

Because of this, frequently used names like 'Nederland' and 'Jansen' have a high probability to be misclassified. The number of 'negative' samples is so large that exceptions like these still happen a lot with respect to the other positive samples and lower precision. This proves that this set of features is not up to the task to find these exceptions.

The last problem is closely related to problem number 2. Using only four features might not be enough to solve this problem; certainly not if the features used are not distinctive enough. Having more features enlarges the chance that a combination of features is very good at classifying positive and negative samples.

## *7.5 Improving the Baseline method*

Since the use of the features selected in section 7.3 did not improve the baseline method a new approach was needed. One of the problems was that when all possible combinations of clusters are compared it is 'inevitable' that clusters are matched erroneously (at least with the information currently available). A possibility is not to compare all the clusters but only clusters that have a high probability of being co-referential, like the baseline method does. The drawback of such a method is that it is unlikely to find the hard cases; cases with very different names.

Instead of just matching individual names as the baseline does it is possible to match the names in the clusters provided by the single document NE co-reference resolution. In this way the knowledge about the different names that co-refer can also be used for cross-document methods. The rule that was used to do this is the following:
*Two clusters of names are co-referential if the most frequent name in cluster 1 is the same as the most frequent name in cluster 2. (If there are more candidates for the 'most frequent name' the name containing the most words is used)*

The intuition behind this rule is very simple, if a NE is referred to most frequently using name A in $document_1$ and the same name is used most frequently in $document_2$ they are likely to be co-referential.

### 7.5.1 Results

The evaluation was done using the exact same methodology and the same 215 documents as used on the baseline method. The results from both the baseline and the 'more advanced method' are shown in the table 7.13 below.

| Evaluation metric | Baseline method | Advanced method |
|---|---|---|
| Precision | 0.86 | 0.82 |
| Recall | 0.67 | 0.74 |
| F-measure | 0.75 | 0.78 |
| #clusters 100% correct | 21% | 37% |
| #clusters 100% precision | 70% | 58% |
| #clusters 100% recall | 30% | 49% |

**Table 7.13**: results of the more advanced cross-document NE co-reference resolution method in comparison to the baseline method

This method has roughly the same effect as the advanced method used in the single-document step; precision is lower and recall is higher. This is also reflected by the percentages of clusters with perfect precision or recall. The percentage of clusters that is 100% correct was raised from 21% to 37% percent which is a significant increase.

### 7.5.2 Discussion of the results

Looking at the F-measure this method performs slightly better than the baseline method. The main reason for this is the higher recall score which apparently fixed a lot of the existing mistakes and raised the number of clusters that is 100% correct. This method has several strong and weak points.

Strong points:

- This method is fast; comparisons can be done on the database.
- No extra information extraction (like context) needs to be done (also improving speed)

Weak points:

- Lower precision
- Co-referential names that that have different surface forms are not found
- Names that do not co-refer to the same entity but have the same surface form have a high chance of being grouped together. This is a bigger problem than in the single-document method since the singe sense per discourse principle is not valid for multiple documents (there is no single discourse).

The result is far from perfect and there is a lot of room for improvement. Unfortunately there was not enough time to further improve this method and thus it was used to evaluate the impact of co-reference resolution on the text mining task.

## *7.6 Conclusion*

This chapter described the work done to solve cross-document NE co-references, including the hard cases where names that co-referential have very different surface forms (or the other way around). The assumption was that all the clusters need to be compared in a smart way in order to find all the co-referencing clusters. To do this a number of promising similarity features were defined to compare clusters found by the JW-method:

1. Name similarity
2. Document-context similarity
3. Co-occurring names similarity
4. Difference in publish dates

Three different machine learning techniques were tried to train a good classifier on these features. Unfortunately none of the classifiers performed well with best F-measure of 0.59. An important reason for this 'failure' is the large number of comparisons between non-co-referential clusters. Even if the chance of determining such clusters as co-referential is very small the large number of comparisons done will still result in a lot of mistakes.

Another reason for the bad performance of the classifiers has to do with the number and the distinctiveness of the features. The current method has four features which is very little for most machine learning algorithms. With more features there is a better chance of learning a model that is capable of 'recognizing' exceptions.

Multiple machine learning techniques were trained on different sets of features. The results show that feature 1 is by far the 'strongest' feature (getting the highest weights) and that the other features are not important. This means that features 2, 3 and 4 can not correct mistakes made as a result of feature 1 (the exceptions).
There can be several causes for this problem:

1. The features are not calculated correctly or must be calculated in a different way so they get more meaning.
2. Features 2, 3 and 4 are all extracted from only 1 document because that is the starting point of the cross-document method (single documents with clusters of names). It is possible that when these features are extracted from multiple documents they are stronger (especially features 2 and 3).

The baseline method defined at the start of this chapter omits these problems by only using name similarity and not comparing all cluster combinations. A method with slightly higher F-measure (due to lower precision but higher recall) was developed which only uses name similarity. This method uses no extra information and is very fast. However, the drawback of this method is that is fails to solve the hard cases. Unfortunately there was no time to improve this method and hence this method was used to assess the influence of NE co-reference resolution on text mining.

A possible solution can be to only classify clusters of names that originate from documents that are related in some way (by for example using labels from AdjustServlet). Using this strategy the number of 'negative samples' will be a lot less resulting in better balanced data. Another advantage of this method is that it lowers the total number of comparisons done resulting in better scalability.
However this method has the risk that if documents are not related but contain the same NE important relations for this NE can be missed. A simple example is a person A that is active in both politics and sports. The political-articles are not related to sports-articles and thus these two 'sides' of person A will not be found. In the end this method will result in two different person A's both covering a part of the relations.

# 8 Impact of named entity resolution on text-mining

## *8.1 Introduction*

The goal of this research is to assess the impact of cross-document co-reference resolution on text mining. In the case of this research the mining of 'relations' between named entities done by Novalink. The goal is to find out if the results form Novalink with NE co-reference resolution are better (or worse) than the results found by the 'old' Novalink and in what way. This chapter describes how this evaluation was done and presents the results.

## *8.2 Evaluation Methodology*

The requirements for the evaluation of the text mining are largely the same as the ones used to choose the (cross) document NE co-reference resolution evaluation method. These requirements are:

1. Give a good indication of the quality of the text mining results
   The results are good (or better than other results) when:
   - the results contain less doubles
   - the results has more correct relations
   - the relations found have a better order
2. Give a good indication of the strong and weak points of the results
3. Give a clear indication how much better or worse the 'new' results are compared to the 'old' results.
4. Be understandable for other computer scientists active in related research-fields

In this evaluation the results from a text mining task using cross-document NE co-reference resolution are compared with the same task that does not have this extra information. Novalink was used as text mining tool to evaluate this.
It is not easy to evaluate the new Novalink (called Novalink-NECR) in a quantitative way since there is no ground truth data available of the relations. The only other option is to compare the results from Novalink-NECR with the results from Novalink (both based on the same documents). As described before, Novalink generates a graph with named entities as nodes and relations as edges. Each edge has a score denoting the strength of the relation, the higher the number the stronger the relation. This graph can be seen as an ontology with specific knowledge about NE's. The graphs from Novalink and Novalink-NECR can be evaluated and compared using methodology from ontologies.
There are multiple ways to evaluate ontology's. An overview of these methods is given by Brank et al. (Brank, Grobelnik, Mladenić; 2005) who describe four different evaluation approaches:

1. Comparing the ontology with a ground truth model
2. Evaluate the results of an application that uses the ontology
3. Compare the ontology with data about the domain that is covered by the ontology
4. Evaluate the ontology using predefined criteria, done by humans

Approaches 1 and 2 require ground truth data for either the ontology or for the application. There is not ground truth data available for Novalink and the creation of this data is very time consuming. There was no time available to create ground truth data and hence the first two evaluation approaches could not be used. The problem with the 3rd option is that the graphs do not really cover a specific domain, at least not one that is literally described in a set of documents. This means that approach 4 is the only remaining option.
According to Brank et al. the standard method in approach 4 is to give a score for each pre-defined criterion, the overall score is then a weighted sum of these individual scores.

A small preliminary test was done to assess this way of evaluation on the results of Novalink. The heuristics used all had to do with scoring some aspect of the graphs (number between -1...1):

- Number of doubles in both results
- The correctness of the relations found by Novalink-NECR that were not found by Novalink
- The correctness of relations found by Novalink that were not found by Novalink-NECR
- The order of the relations

Four sets of graphs were evaluated using these heuristics and this method was found to be very hard. Even though the differences between the results of the two Novalink versions are sometimes very large it is most of the time very hard to tell for each heuristic which one is better. The result is that most scores end up very closely to 0 and fail to tell anything specific (which was the idea behind the heuristics).

An alternative was to take a more user centred approach and evaluate if the new results help to answer the question of the user. The user typically wants an overview of the relations some NE has with other NE's. So if Novalink-NECR gives a better overview then Novalink it is better. In this case 'better' is defined by 2 different things:

- Is the overview correct (are the relations in it correct)
- Is the overview useful with respect to the original search term (does it server its purpose)

Ten people were asked to rate 15 graphs (see next section) with a rating from 1 to 10 based on these two rules. This rating has a large (intuitive) scale so the assessors can rate really precisely. However, people often have different 'base' ratings, where one gives a 6 someone else gives a 7. Fortunately it is not the height that is interesting but the difference in scores the individual assessors give.

These assessors also had to give an argumentation for their rating and could give more general observations. These comments are very valuable since they give an insight into what people expect, want and how this influenced their opinion of the results. Of course the argumentation need to be interpreted and can also give information that is not strictly needed for the evaluation (about visualisation etc). more?

### 8.2.1 Selection of evaluation cases

The evaluated graphs needed to be representative and should also have hard cases in order to give a good indication of the performance of Novalink-NECR. The assessors need to know the context of the different cases; else they can not rate them. This 'limited' the choice to well known people, organizations and locations. The following 15 cases were used:

- Verdonk
- Marco Borsato
- Arena
- Holleeder
- Schiphol
- Albert Heijn
- Jaap Stam
- Zweden
- Condoleezza Rice
- Merkel
- Alpen
- Bin Laden
- OPEC
- Zuid-Afrika
- Fortuyn

There is a variety of people (8), locations (4) and organizations (3) as well as Dutch and non-Dutch names. The names also differ from the field they come from: music, sport and politics. Most names originate from politics because they are well known which makes them easier to evaluate. The different Novalink results for these 15 cases (and their average ratings) can be found in Appendix D.

## *8.3 Results*

The average scores are presented in table 8.1. The first column is the total average; the subsequent columns are the average scores for the three different types. The standard deviation for the ratings for Novalink results is 0.97, the standard deviation for the ratings given for Novalink-NECR is 1.02.

| | **Total** | **Persons** | **Locations** | **Organizations** |
|---|---|---|---|---|
| **Novalink** | 6.68 | 6.69 | 6.73 | 6.59 |
| **Novalink-NECR** | 6.74 | 7.01 | 6.18 | 6.78 |

**Table 8.1:** average scores found in the evaluation

In some cases the ratings for Novalink and Novalink-NECR differ a lot (up to 4 points). This really shows the occasional dissatisfaction for one of the results. Another important observation is the difference between the assessors, there are very few cases where the assessors agree that one is better than the other. This fact, in combination with the provided argumentation indicates that the assessors have different expectations and find different aspects of the graphs important.

The following part is split into two sections: observations and assessor comments. Observations are things that can 'simply' be seen in the graphs and were sometimes mentioned by assessors. The comments section contains frequent argumentations used to explain ratings.

Important observations:

- Novalink-NECR has less double entities than Novalink. In total the results from Novalink contained 31 doubles which is 14% of all entities. Results from Novalink-NECR had 11 doubles which is 3% of total.
- Novalink-NECR found 25% more relations than Novalink (Novalink found 219 relations while Novalink-NECR found 292). This is of course closely related to the number of doubles, less doubles means more space for other relations.

- Novalink-NECR is more dependent on the search term used than the old Novalink version (this was found when creating the evaluation). For Novalink-NECR it is important to use the name of the entity that is most common used in the news. For example if you want information about Bush you should search on 'Bush' and not on the full name (George Bush, G.W. Bush). This is less important for the old system since it did not do single-document co-reference resolution. The reason is described in more detail in the discussion section.

Important comments:

- The results from Novalink-NECR sometimes go out of the expected 'domain'. When looking at the initial search term the assessors assume a certain domain (politics, music, sports etc). When there are relations outside of this domain (or when it is unclear) assessors doubt the validity of the relations and give lower ratings. For example one of the relations found for 'Marco Borsato' (a musician) is a 'Nijs' who has political relations, which is very strange. This happens more in the results of Novalink-NECR than in the results of Novalink.
- Some of the results found by Novalink were 'unexpected', like the relation 'Geert Wilders' – 'Bill Clinton'. This happens more using Novalink-NECR than in Novalink and a lot of people palatalized this heavily. Novalink-NECR also shows more relations which enlarges the probability on mistakes. But this does not fully explain the number of mistakes made by the Novalink-NECR.
- Novalink-NECR displays more relations and often these relations are more helpful than relations found by Novalink (which sometimes only displays very few entities with a lot of relations).
- One of the assessors stated his view on the matter: "Novalink-NECR displays more relations which makes it better than the other results. I can always check the relations to find out if they are wrong, but I can not investigate relations that are not displayed at all.".
- Most of the assessors indicated the difficulty of the task, it was often more difficult and time consuming than they anticipated. This indicates that even though the results are often very different it is hard to tell which of the results (if any) is really better.

## *8.4 Interpretation and explanation of the results*

The difference in scores of the two systems is very small, apparently the assessors did not find one system particularly better than the other. A remarkable observation is that where the results for persons and organizations are rated slightly higher for Novalink-NECR, locations are rated a lot lower. The related entities found by Novalink-NECR were often not expected and or not really informative.

Even though the average ratings are very similar the graphs are often very different. The most important differences are:

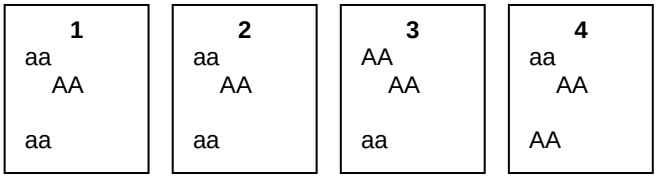
- Novalink-NECR has less doubles and as a result displays more relations
- Novalink-NECR displays more relations that are unclear, go outside of the expected domain or are wrong.

The rest of this section will explain where these differences come from.

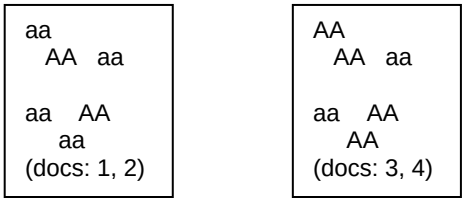
One of the things that explain these results is the way the text mining is done. The text mining is based on documents and whether two names occur within the same document or not. To get good results for an entity you need to find all the documents this entity occurs in (independent what names are

used to refer to this entity). Whether multiple names for the entity occur within the document does not matter.
The cross-document NE co-reference resolution method groups the documents together based on clusters of names found by the single document method. If this clustering is not done correctly there is a high chance to miss documents and hence possibly miss relations. This does not happen in the old method since it does not perform single-document co-reference resolution. Below is an example to explain this.

**Example:**
An entity A has two name variants: aa and AA. There are in total 4 documents that contain these names:

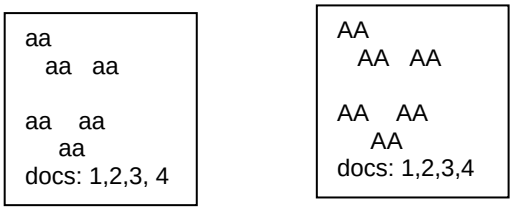



The single-document method extracts clusters of names from these documents and matches the AA and aa instances in each document together. Then these 4 small clusters of names are compared in the 'cross-document step' using the most common names in each cluster. The A clusters from documents 1 and 2 are grouped together because they both share the common name 'aa'. The same goes for the clusters extracted from documents 3 and 4 who share 'AA' as common name. This results in the following two clusters:




This is of course a mistake of the cross-document step which should have matched all these clusters together. The results for text mining on entity A (using search term aa or AA) results in the uses of either documents 1, 2 or documents 3, 4 and hence missing information from the other two documents!

The baseline method of the 'old' Novalink also creates two clusters of names but these are different from the ones above:




Even though these clusters are far from perfect they result in the usage of all four documents for text mining.
This also explains where the doubles in the old results often come from. When for example the 'aa' cluster is used to find relations for A it is very likely that 'AA' is found with a very high co-occurrence (because no single-document co-reference resolution was done). This does no longer happen in Novalink-NECR

This example is very extreme but not far beside the truth which was observed in the database. This does not happen for all names but happens for names that have multiple 'standard' variations. An example is Willem Holleeder. In some documents he is being referred using his full name while in other documents mostly the last name is used. The cross-document module then erroneously assumes that these names are not co-referential. This is what happens in the example.
But other names are used more consistent in all documents, like the name of the Dutch Prime Minister Balkenende. This name also has variations but still 90% of all occurrences in all documents just use the last name. In this case the problem described above does not occur.

This 'phenomenon' explains a lot of the results:

- A clear difference between the old and new results is the lower number of doubles. The reason for this is the single-document step which finds these names within single documents.
- The big differences between results. Some results are better than the old method while other results are a lot worse. In the case of worse results it is mostly the case that this phenomenon occurred and not all the documents were used for text mining while the 'old' method did use all (or at least more) documents.

An important and unexpected conclusion can be extracted from this knowledge: *High recall scores for the clustered names do NOT directly indicate better text mining results (as assumed throughout this research).*

If the names from a cluster with a low recall score occur in most of the important documents the text mining will perform very well (it doesn't care that 3 from 4 co-referential names within a single-document are not found). And the opposite is also true; a cluster with a relatively high recall score (due to names it found in single documents) can have members from a select number of documents which result in 'bad' text mining results.
In general the assumption that co-reference resolution done with a high recall score results in better text mining is correct. However it does not ensure valid text mining results, it largely depends on the entity and its name-variations someone is looking for.

The NE co-reference resolution precision of Novalink-NECR is lower than the precision of Novalink. This means clustering names that are not co-referential. This can result in erroneous relations. An example of this is the earlier mentioned relation between 'Marco Borsato' and 'Nijs'. This 'Nijs' appears to be 'Annette Nijs' who is a politician but does not have a relation with Borsato. It is more likely that Borsato has a relation with another musician called 'Rob de Nijs' who was erroneously put together with the politician. Apparently Annete Nijs was a lot more in the news resulting in the 'political' relations which a lot of the assessors found very odd and palatalized. This also explains why the results from Novalink-NECR sometimes go outside of the 'assumed domain'. It is very likely that this also occurs within Novalink but due to higher precision and in general the display of 'fewer' relations it is far less obvious.

## 8.5 Conclusion

The most important conclusion that can be extracted from these results is that neither system is better than the other; they both get the same ratings on average. From this point of view the NE co-reference resolution done did not really work because it did not result in better text mining results. However, the results from the two systems were sometimes very different which resulted in some interesting findings. These findings and some 'important lessons learned' are described below.

The 'doubles problem' was largely solved by the NE co-reference resolution method used. There can still be double entities within the results but it is very unlikely that these are direct children of each other. The fact that these doubles are gone is a result of the single-document NE co-reference resolution method.

High recall of the co-referential names does not necessarily mean better or more text mining results. In the case of Novalink this has to do with the way relations are calculated. For Novalink it is more important to find all the documents a name occurs in than to find *all* the co-referential names. In general high recall of co-referential names is an indication that most of the documents are found. But a high recall can come from relatively few documents which have a lot of names instead of having all the documents the name occurs in. For a better understanding of this the reader is encouraged to read the example in section 8.4.
This means that the assumption used throughout this research that a *higher recall for co-referential names automatically results in better text mining* is not fully correct. It is more important to find all the documents a certain NE is mentioned in.

The lower precision score of the NE co-reference resolution method used with respect to the baseline method is responsible for more mistakes in the results of Novalink-NECR. That Novalink-NECR displays more faulty relations/entities is also a result from the simple fact that Novalink-NECR displays more relations (due to less doubles). The assessors indicated that these mistakes, or strange results are often not desirable. This indicates that *the aim to get higher recall for co-referential names at the cost of precision should be used very carefully*.

The strongest part of the NE co-reference resolution method used to pre-process the data for Novalink is the single-document step. However, the (generally accurate) information found in this step is not used by Novalink to calculate relations. As denoted before it is all about 'finding the right documents' and not about the 'content' of each document. A possibility can be to use the information of the single-document NE co-reference resolution for the calculation of these relations. An intuitive option is to use the number of names in a document that refer to the entity someone is interested in as a weight for the relations found in that document. If for example the user is interested in entity A, the relations found in documents where A is mentioned a lot can be found to be more important than relations found in documents where A is mentioned only once. This is of course an intuitive statement and should be tested in future work.

# 9 Discussion

This discussion section will look back at the work done and point out the strong and weak points and how they can be improved. The different aspects of this research will be discussed in separate sections.

## *9.1 Ground truth data*

The ground truth data, in the form of an annotated corpus, had a lot of influence on the research. It was used for testing, tuning and evaluations. A final check was done to ensure cross-document NE annotations were correct. The corpus was made as fine-grained as possible, meaning that names that refer to different aspects of the same thing are considered to not co-refer. A commonly used example of this phenomenon are location names, such as Nederland, which can refer to a lot of different things. Semantically this is correct, but the question is if the distinctions made are meaningful for the text mining task. The various NE's can also be viewed as multiple metonymical interpretations of the name of one geographical (for example the government, military, inhabitants of Nederland).
In the current NE co-reference resolution system names like these are erroneously mapped together resulting in precision errors. These mistakes are very hard to correct and this raises the question whether they are needed in the first place. The same question also concerns non-rigid designators. Are they important enough to solve, or is their contribution to the final result so minimal that they can be omitted? Answering these questions in an early stage of the research would have given more insight into the problem and could have lowered the 'complexity' of the problem.

## *9.2 Solution strategy*

A bottom-up strategy was used to solve the problem (single document before cross document NE co-reference resolution). Another option is to solve the mistakes made within Novalink i.e. the doubles in the graphs. In that case expensive pre-processing is not needed and two related entities with similar names can be clustered together very easily. In this way synonymic names can be fixed (homonymic names can not be fixed in this way).
However, some NE co-reference resolution is always needed to be able to calculate relations in the first place. It might be possible to correct some of the mistakes made afterwards but this 'top down' strategy does not tackle the real problem.

## *9.3 NE co-reference resolution evaluation methodology*

A lot of time was used to find / develop a good evaluation methodology to assess the quality of different NE co-reference resolution methods. The methodology selected in the end is capable to calculate precision and recall in a fair manner and each 'mistake' is treated in a fair way. Altogether the evaluation methodology used gives a good insight into the performance of evaluated methods.
It might be possible to extend the evaluation methodology with a distinction of easy and hard cases (analogue to Mitkov's trivial and non trivial accuracy). This would give more insight into the quality of evaluated methods.

## 9.4 Single document NE co-reference resolution

An important part of this research was the study into single-document NE co-reference resolution. Three 'name similarity' based systems were developed in an iterative way. The best system perform pretty well; it can handle most common Dutch name structures and spelling variations. It should be noted that part of this method was specially designed to cope with NNER-mistakes. Altogether this study provides a lot of insight into the different forms names can have and how to compare them meaningfully. I do not think the proposed method can be improved a lot using only string similarity (also see future work).

## 9.5 Cross document NE co-reference resolution

The cross-document NE co-reference resolution task turned out to be very hard. The large number of names, exceptions and NNER-mistakes make it difficult to solve co-referring NE's with high certainty. Hence there are a lot of aspects of the cross-document NE co-reference resolution method have room for improvement or can be done differently:

- The number of features is rather small making it hard for any machine learning method to train to train a good model. A larger number of features give a higher chance to build classifiers with better performance (if the features are good enough).
- The similarity of features can be calculated in a different way which might be more valuable for machine learning methods to train on.
- The cross-document NE co-reference resolution strategy was based on the idea that all clusters need to be compared in order to find all the co-referential clusters. However this methodology is very error-prone and time consuming. Even with a very small chance on a mistake the large number of comparisons results in a 'large' number of mistakes. Other strategies can be used to diminish this problem (for example first cluster the documents and then do NE co-reference resolution within these clusters).

Even though the final NE co-reference resolution method is only slightly better than the baseline method it influenced the text mining results quite a bit and some interesting conclusions were be drawn from the results.

## 9.6 Complexity of NE co-reference resolution methods

Due to time constraints the speed and complexity of NE co-reference resolution methods got minimal attention. One section describes the complexity of the single-document methods, but this was done afterwards and the knowledge was not used as selection criterion.
This research was more concerned with the development of accurate co-reference resolution methods than with the optimization of these methods. The main reason for this was the lack of time. Still the complexity is an important aspect of NE co-reference resolution research and should not be taken lightly.

## 9.7 Evaluation of Text mining

It difficult to evaluate the influence of NE co-reference resolution on text mining since there was no ground truth data available. The evaluation of Novalink was subjective and a lot of other aspects beside the influence of NE co-reference resolution were included in the evaluation (like the visualisation of the relations). There were some things that could have influenced the results or could have done differently:

- People that know one of the two Novalink versions could tell which graphs were created by which system because of specific colouring of the nodes in the graphs. This could have caused a bias toward either system. However, it is likely that without the colouring these people would still have recognized the systems because of other differences (like the lack of doubles etc).
- The information provided about the graphs was a bit minimal which raised questions (what does the colour of the arrows mean, why is it structured the way it is etc.). In some occasions this lack of information confused the assessors; it was sometimes unclear what they were looking at.
- The evaluation was setup in a way that the assessors could write down everything they thought. This methodology provided a lot of information, unfortunately this information was not always useful with respect to the purpose of the evaluation. An good alternative would have been to use more directed questions.

# 10 Conclusion

The research described in this thesis was done to improve name based text mining by solving cross-document named entity co-references. The 'old' Novalink string matching based NE co-reference resolution method served as the baseline for this research. A cross-document NE co-reference resolution method was developed which has a higher recall but lower precision (and slightly higher F-measure) than this baseline method. This solution was implemented in Novalink and the text mining results were compared with results from the original Novalink version. The results and observations from this evaluation together with results from the studies into single- and cross-document NE co-reference resolution methods are used to answer the research questions described in chapter 1.

## *10.1 Main research question*

The main research question was:
*Can Dutch text mining be improved using named entity co-reference resolution?*

The NE co-reference resolution method used largely solved the 'double-problem'. The F-measure of this method is not a lot higher than the F-measure of the baseline method. This indicates that NE co-reference resolution has a lot of impact on text mining and that small changes can already have a lot of influence. Even if methods are do not outperform each other (looking at precision and recall) the clusters created can be very different and as a result the text mining results can differ a lot.
The method developed in this research is very different from this baseline method and the evaluation did indicate some interesting aspects (answering sub-question 3):

- Single document NE co-reference resolution is needed to omit doubles in the Novalink results.
- By lowering the number of doubles there was more room for other entities which resulted in the average display of 25% more relations.
- The NE co-reference resolution method used has a higher recall (i.e. puts more co-referential names together than the baseline method). However, the results indicate that this does not always result in better (finding more relations with less 'mistakes') text mining results. A small representative group of names can result in valid text mining results. But in general larger recall does indicate 'better' text mining results.
- A high precision of co-referential names is very important because low precision leads to strange and unexpected results. The evaluation indicates that people really dislike this. In addition, the presence of erroneous relations sometimes leads to lower confidence in the entire network.

## *10.2 Sub questions*

The first and second sub-questions are about information (features) that can be used for NE co-reference resolution. This research was split in two parts to answer these questions: single- and cross-document NE co-reference resolution.
The single-document study focused on the usability of name surface form similarity for names that occur within the same document. This study found that name similarity is an important piece of information to conclude whether two names are co-referential. The similarity assessment of two names must be done carefully. Normalization is always needed to filter out anomalies or language specific constructions. The study also found that it is best to use string edit distance metrics on the individual words that make up names than to compare entire names.
However, string similarity methods are not perfect and can not handle cases where two names are either homonymic or are very different but co-referential. More knowledge (syntactic or semantic) is needed to solve these cases.

The cross document NE co-reference study focused on the use of additional features:

1. Name similarity
2. Document-context similarity
3. Co-occurring names similarity
4. Difference in publish dates

These features were used to train models capable of distinguishing co-referential names from non-co-referential names. Unfortunately none of these models proved good enough, relatively much non-co-referential names were clustered together resulting in low precision. The reasons are described elaborately in chapter 7.
This does not directly mean that these features are useless for NE co-reference resolution (very similar features are used with success in literature). These features also seem logical to use; a human that must do NE co-reference resolution could probably use these features. More research is needed to see how features like these can be used properly (also see the discussion chapter regarding this issue).

In the end a relatively simple cross-document NE co-reference resolution method was used with reasonable results. This indicates that name similarity is also valuable in cross-document context. However, it is not likely that methods solely based on name similarity will be able to do cross-document NE co-reference resolution with high accuracy. Information like the features described above will be needed but must be used carefully.

The NE co-reference resolution method developed in this research also has potential that was not utilized by Novalink. Especially the results from the single-document method can be used in a number of ways:

- The method used to calculate the relations can now use the knowledge about the number of names for one entity that is used within a single document.
- The system can point out more passages within documents that concern the entity the user is interested in (a function of Novalink that was not yet mentioned before).
- The different names of one NE can be displayed in the graph instead of one (often not very useful) name to get a better idea about the entity.

# 11 Future work

This chapter will describe possible directions to improve and explore NE co-reference resolution. These 'directions' described here are based on experience gained during this research. The different sections are ordered according to their importance.

## *11.1 Explore the possibilities of syntactic and semantic knowledge*

This research has shown the power but also the limitations of solely name-surface-form based methods. It will be hard to significantly improve the results without using other kinds of information. So the most important focus of future work should be on the usage of more syntactic and semantic information to solve NE co-reference resolutions. This kind of information could be useful for both single- and cross-document resolution. One possibility is usage of name-type information as described in chapter 6.5. If it is possible to determine the type of a name with high accuracy it can be used very straightforward for NE co-reference resolution.

## *11.2 Cross-document NE co-reference resolution strategy*

A compare-everything strategy was used for cross-document NE co-reference resolution. This strategy has a great number of disadvantages which are described in chapters 7 and 9. It is very important to explore different strategies to omit these problems and enhance both the complexity and quality of NE co-reference resolution systems. In my opinion there are a number of options that can be explored (note that this list is just an indication and thus not necessarily complete):

- First do clustering on the documents before performing NE co-reference resolution. This would lower the complexity and it would raise the change that names (with similar spelling) are co-referential.
- Only use a selection of the names found instead of all the names. A lot of names within a document set are so rare that it is unlikely that they will be used for text mining. It might be possible to assess the importance of a name and conclude if the name can be removed or not. One possibility is to look at the number of times a name is used within a document (is it only once than remove it). In this way the number of names in the system is lowered drastically which lowers the complexity.

## *11.3 Improving the named entity recognizer*

Since an important part of any co-reference resolution system will be based on the surface forms of the names it is important that the names are recognized correctly. From this point of view the NNER used in this research should be improved to give better results. One possibility is a post-process to check and correct recognized names using other names in the document. This is an easy and straightforward solution that might proof very effective.

### 11.4 Explore ways to find and fix mistakes in a post-process

It is mentioned before that the strategy used now is bottom-up. There must always be a good bottom-up method to find co-referencing NE's in order to be able to do text mining. However, even a good method can make mistakes that become visible after text mining (like double entities in a result graph). It is probably relatively easy to find and mistakes like these in the text mining results. It might be possible that also other checks can be done after the NE co-reference resolution is done to find and correct mistakes.

# 12 References


Agarwal R.; 1995; Evaluation of Semantic Clusters; *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*; pages 284-286

Bagga A., Baldwin B.; 1998; Entity-based cross-document coreferencing using the Vector Space Model; *Proceedings of the 17th international conference on Computational linguistics - Volume 1*; pages 79-85

Bagga A., Baldwin B.; 1998; Algorithms for Scoring Coreferene Chains; *Proceedings of the Linguistic Coreference Workshop at the First International Conference on Language Resources and Evaluation*; pages 79-85

Cohen J., 1968. Weighted Kappa: nominal scale agreement with provision for scaled disagreement or partial credit; *Psycholog. Bull* - 70, pages 213–220

Cunningham H., Maynard D., Bontcheva K., Tablan V.; 2002; GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications; *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*

Daelemans W., Van den Bosch, A.; 2005; Memory-Based Language Processing (book); Cambridge, UK: Cambridge University Pres

Dunning T.; 1993; Accurate Methods for the Statistics of Surprise and Coincidence*; Computational Linguistics - Volume 19*; pages 61-74

Brank J., Grobelnik M., Mladenić D.; 2005; A Survey of Ontology Evaluation Techniques; *Conference on Data mining and Data Warehouses*;

Bunescu R., Pasca, M.; 2002; Using Encyclopedic Knowledge for Named Entity Disambiguation; *11th Conference of the European Chapter of the Association for Computational Linguistics*

Branting L.K.; 2003; A Comparative Evaluation of Name-Matching Algorithms; *Proceedings of the 9th international conference on Artificial intelligence and law ICAIL '03*; pages 224-232

Chang C., Lin C.; 2001; LIBSVM : a library for support vector machines; Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chang C., Lin C.; 2007; A Practical Guide to Support Vector Classification

Cohen W.W., Ravikumar P., Fiendberg S.E.; 2003; A Comparison of String Distance Metrics for Name-Matching Tasks; *18th International Joint Conference on Artificial Intelligence, Workshop on Information Integration on the Web*;

Fleischman M., Hovy E.; 2002; Fine grained classification of named entities; *Proceedings of the 19th International Conference on Computational Linguistics – Volume 1*; pages 1-7

Fayyad U.M.; Piatetsky-Shapiro G., Smyth, P.; 1996; From Data Mining to Knowledge Discovery: An Overview. *In Advances in Knowledge Discovery and Data Mining*; pages 1–30

Florian R.; 2002; Named entity recognition as a house of cards: classifier stacking; *International Conference On Computational Linguistics, Proceeding of the 6th conference on Natural language learning - Volume 20*; pages 2-4

Florian R., Ittycheriah A., Jing H., Zhang T.; 2003; Named Entity Recognition through Classifier Combination; *Proceedings of CoNLL-2003.*

Gale W.A., Church K.W., Yarowsky D.; 1999; One Sense per Discourse. *Proceedings of the 4th DARPA Speech and NaturalLanguage Workshop*. pages 233-237

Grishman R.; 1994; Wither Written Language Evaluation?; *Proceedings of the Human Language Technology Workshop*; pages 120-125

Hearst M.A.; 1999; Untangling text data mining; *Proceedings of ACL'99: the 37th annual meeting of the association for computational linguistics;* University of Maryland

Hobbs J.;1978 'Resolving pronoun references'. Lingua, 44, pages: 339-352.

Hulse J. (van) , Khoshgoftaar T.M., Napolitano A.; 2007; Experimental Perspectives on Learning from Imbalanced Data; *Proceedings of the 24th International Conference on Machine Learning;* pages 935-942

Jaro M. A.; 1989; Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida; *Journal of the American Statistical Association 89*; pages 414-420

Jurafsky D., Martin J.H.; 2000; Speech and Language Processing: an introduction to Natural Language Processing; *Computational Linguistics and Speech Recognition*; Prentice-Hall inc.

Kim J., Kang I., Choi K.; 2002; Unsupervised named entity classification models and their ensembles; *Proceedings of the 19th international conference on Computational Linguistics - Volume 1*; pages 7-7;

Lee D., On B., Kang J., Park S.; 2005; Effective and scalable solutions for mixed and split citation problems in digital libraries; *Proceedings of the 2nd international workshop on Information quality in information systems*; pages 69-76;

Li H., Srihari R.K., Niu C., Li W.; 2003; InfoXtract location normalization: a hybrid approach to geographic references in information extraction; *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references - Volume 1*; pages 39-44;

Linguistic Data Consortium; 2005; ACE English Annotation Guidelines for Entities; http://projects.ldc.upenn.edu/ace/docs

Louis A., de Waal A., Venter C.; 2006; Named entity recognition in a South African context; *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*; pages 170-179

Mitkov, R.; 2002; Anaphora Resolution; Pearson Education Limited

Niu C., Li W., Srihari R.K.; 2004; Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction; *Proceedings of the 42nd Annual Meeting on Association of Computational Linguistics*; article no. 597

Nenkova A., McKeown K.; 2003; References to Named Entities: a Corpus Study; *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003--short papers - Volume 2*. pages: 70-72;

Peng Y., He D., Mao M.; 2006; Geographic Named Entity Disambiguation with Automatic Profile Generation; *IEEE/WIC/ACM International Conference on Web Intelligence*; pages 522-525;

Ramshaw L.A., Marcus M.P.; 1995; Text Chunking Using Transformation-Based Learning; *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94

Vilain M., Burger J., Aberdeen J., Connolly D., Hirschman L.; 1995; A model-theoretic coreference scoring scheme; *Proceedings of the 6th conference on message understanding (MUC-6)*; pages 45-52

Wakao T., Gauzauskas R., Wilks Y.; 1996; Evaluation of an algorithm for the recognition and classification of proper names; *Proceedings of the 16th conference on Computational linguistics Volume 1*; pages 418-424;

Wacholder N., Ravin Y., Choi M.; 1997; Disambiguation of Proper Names in Text; *Proceedings of the 5th Conference on Applied Natural Language Processing*; pages 202-208;

Yarowsky D.; 1995; Unsupervised Word Sense Disambiguation Rivaling Supervised Methods; *Proceedings of the 33rd Annual Meeting of Association for Computational Linguistics*

Yang Y., Yoo S., Zhang J., Kisiel B.; 2005; Robustness of Adaptive Filtering Methods In a Cross-benchmark Evaluation; *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR '05;* pages 98-105

Zhang J.; Mani I.;2003; kNN Approach to Unbalanced Data Distributions: A Case Study involving Information Extraction; *Proceedings of The Twentieth International Conference on Machine Learning (ICML-2003), Workshop on Learning from Imbalanced Data Sets II*

# Appendix A

## Handleiding

### Het doel van het annotatie werk

Voor mijn afstudeer opdracht moet ik een methode/algoritme ontwikkelen dat bepaald welke 'entiteit' (persoon, organisatie etc) in de wereld hoort bij een naam in een tekst. Hierbij gaat het dus om verschillende namen die naar dezelfde entiteit verwijzen of om dezelfde namen die naar verschillende dingen verwijzen. Om een methode te kunnen ontwikkelen die dit doet zijn documenten nodig waarvoor dit al is opgelost. Met behulp van een dergelijke set documenten kunnen methodes worden getest, ge-finetuned en worden geëvalueerd. Het doel van dit annotatie werk is dan ook om een dergelijke set documenten te maken. In totaal gaat het om ongeveer 300 documenten waarvoor voor iedere naam moet worden aangegeven naar welke entiteit de naam verwijst.

### Installeren GATE (General Architecture for Text Engineering)

Om te beginnen download GATE van http://gate.ac.uk/download/index.html. GATE is een Natural Language Framework dat wordt gebruikt voor het implementeren van NLP modules evenals annoteren van tekst. GATE is een 100% Java applicatie dat zonder administratieve rechten geïnstalleerd kan worden door simpel de setup te draaien en de stappen te volgen. Het maakt niet uit in welke directory GATE geïnstalleerd wordt.
Maak in de GATE-3.1 directory de map 'Corpus' aan en pak hierin het bijgeleverde zip bestand uit. Deze zip file bevat de documenten die door geannoteerd moeten worden.

### Laden van een corpus en annotatie schema

Om documenten te annoteren moeten ze in GATE worden geladen. Documenten kunnen één voor één of per directory worden ingelezen. Aangezien het laden van een hele directory het makkelijkst is zullen wij dat doen. Hiervoor dient eerst een Corpus aangemaakt te worden:

- Selecteer met de rechter muisknop in het linker frame van het venster 'Language Resources' en kies 'New' → 'GATE Corpus'.
- Er verschijnt een popup, druk hier gewoon op 'OK' (het is niet nodig om een naam in te geven).
- Onder 'Language Resources' verschijnt het nieuwe corpus

Nu het corpus is gemaakt moeten er documenten in worden geladen:

- Selecteer met de rechter muisknop het nieuw aangemaakte corpus en kies 'Populate'
- Er verschijnt een popup, druk hier op het 'map' icoontje rechts bovenin het venster om een file browser te openen. Selecteer hier Gate-3.1/Corpus/<de directory die u is aangegeven>. En kruk op 'Open'.
- Druk vervolgens op 'OK'.
- Als het goed is verschijnen alle documenten uit die directory onder het hiervoor aangemaakte corpus.

Vervolgens dient er een annotatieschema geladen te worden waarin de mogelijke annotaties zijn vastgelegd:

- Selecteer met de rechtermuisknop 'Language Resources' en kies voor new → 'Annotation Schema'.
- Er verschijnt een popup, klik hier op de knop met het map-icoon (rechts in het venster).

- Er verschijnt een file browser, selecteer hiermee Gate-3.1/Corpus/schema.xml en druk op 'Open'
- Druk vervolgens op 'OK'.
- In de lijst verschijnt nu een annotatie schema

Opslaan van het Corpus geschied door met de rechtermuis-klik het aangemaakte corpus te selecteren en dan 'Save as XML' te kiezen. Selecteer de juiste folder en druk op 'Select'. Als er al bestanden in die folder bestaan verschijnt er een pop-up die vraagt of de bestanden overschreven dienen te worden, selecteer hier 'all'. Sla de bestanden vaak op, het is jammer als er iets opnieuw gedaan dient te worden! Je kunt er voor kiezen om in de originele directory op te slaan, of om het in een nieuwe directory op te slaan.

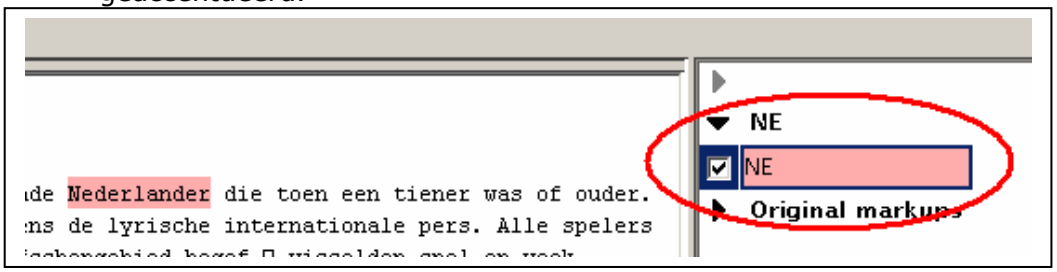
**Annoteren**

Om een document te kunnen annoteren dient simpelweg een document uit het corpus doormiddel van dubbelklik te worden geselecteerd. Als het goed is verschijnt dan de tekst van het document in het middelste scherm. Als de tekst helemaal is doorlopen kan de 'tab' worden gesloten door bovenin met de rechtermuisknop de tab te selecteren en dan 'Hide this view' te kiezen. Kies niet 'Close' (!!) want dan wordt het document zonder het op te slaan uit het corpus verwijderd! Het is ook absoluut niet de bedoeling om de tekst zelf aan te passen, ook al zitten er soms fouten in (missende spaties etc)!

Een GATE-module heeft al geprobeerd alle eigennamen te herkennen en te annoteren. Dit is echter lang niet altijd goed gegaan en deze annotaties moeten worden verbeterd en worden uitgebreid. Verbeter ALLE annotaties waar nodig en voeg nieuwe annotaties toe als Named Entities (NE) zijn gemist

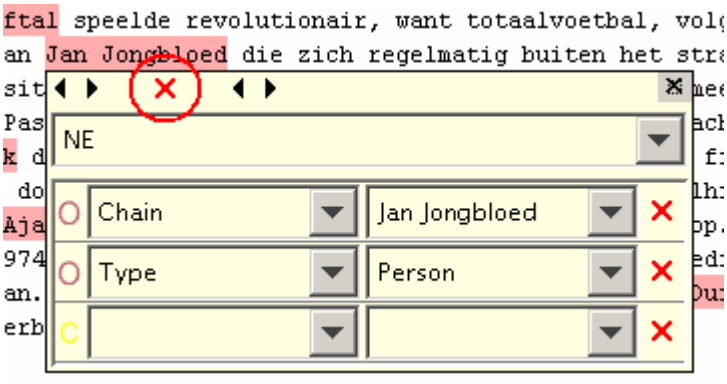
Om deze annotaties te kunnen zien dient bovenin de knop 'Annotation Sets' te worden ingedrukt. Er verschijnt rechts een 'frame' met twee pijltjes.

- Klik op het pijltje waar "NE" bij staat. Er verschijnt een check-box met daarnaast "NE" (dit is de enige annotatie die gebruikt zal worden). Zie de figuur hieronder.
- Selecteer deze check-box, als het goed is worden alle gevonden NE's in de tekst doormiddel van een kleur (waarschijnlijk rood) geaccentueerd.

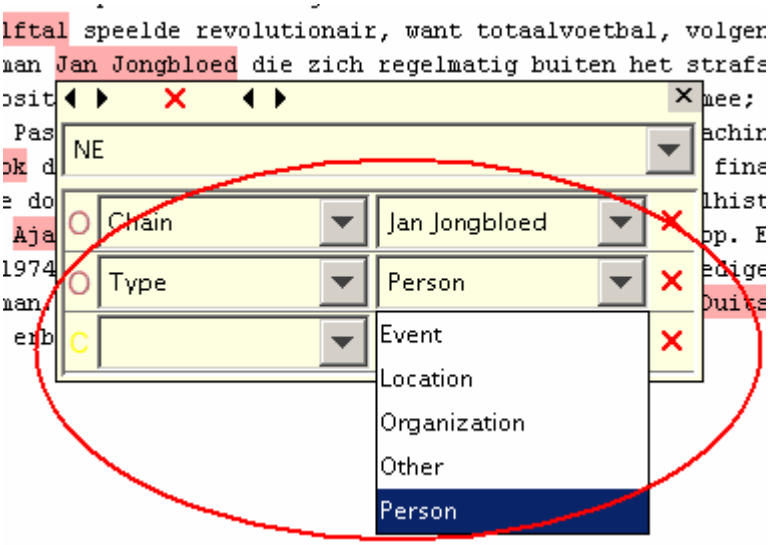



Door met de muis over een annotatie te 'zweven' verschijnt er een pop-upje met daarin de attributen van die annotatie. Door op het GRIJZE kruisje rechts bovenin te klikken verdwijnt het pop-upje. Er zijn een aantal bewerkingen die op een annotatie kunnen worden uitgevoerd:

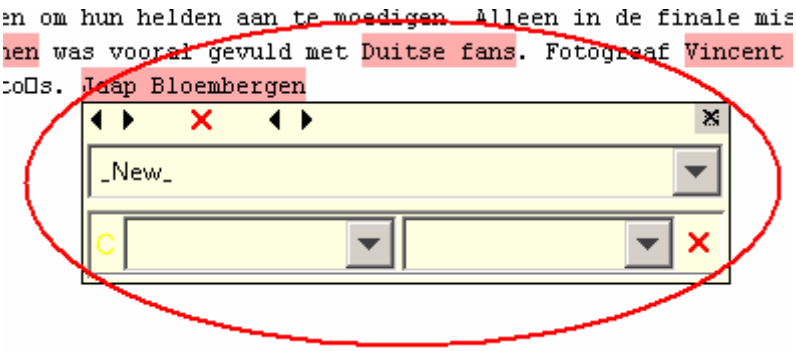
- Verwijderen: verwijder een annotatie door in het pop-upje het RODE kruisje midden-bovenin te gebruiken. Zie figuur hieronder.




- Aanpassen van attributen: dit kan door simpelweg oude waarden te verwijderen en nieuwe te typen of door een mogelijkheid uit het drop-down menu te kiezen. Welke attributen er precies zijn en waar ze voor dienen wordt hieronder uitgelegd. Belangrijk om te weten is dat 'chain' attribuut geen drop down menu heeft en het 'type' attribuut wel!




- Maken van een nieuwe annotatie: selecteer met de muis het stuk tekst dat geannoteerd moet worden en wacht even. Als het goed is verschijnt er een pop-upje met allemaal lege waarden. Als in het bovenste veld nog niet "NE" staat (dan staat er waarschijnlijk "_NEW_") type hier dan "NE" en sluit het venstertje. Door dan nogmaals het pop-upje te openen kunnen de NE-attributen gezet worden.




Iedere eigen naam (NE) heeft twee eigenschappen: de Chain en het Type.

De **chain** geeft aan naar welke entiteit of object in de wereld de desbetreffende NE naar verwijst. De naam 'Holleeder' bijvoorbeeld verwijst naar 'Willem Holleeder' en dit hoort dan ook als waarde bij de chain te worden gegeven. Uiteindelijk is het de bedoeling dat alle NE's in alle documenten die verwijzen naar dezelfde entiteit dezelfde chain-waarde krijgen (alle vormen van holleeder krijgen dan bijvoorbeeld als chain-waarde 'Willem Holleeder'). De GATE-module heeft geprobeerd zo goed mogelijk in te schatten tot welke chain een NE behoort en dit veld is altijd ingevuld. Het is echter goed mogelijk dat deze waarde foutief of onvolledig is en dient dan aangepast te worden. Verbeter deze waarde dus indien nodig, zorg dan wel voor een eenduidige duidelijke naam (of omschrijving). Het is dus erg belangrijk dat eerdere en latere voorkomens dan die naam precies dezelfde waarde krijgen! Zie ook de voorbeelden achteraan dit document.

Vaak komen dezelfde namen veel voor in een document. Het komt dan ook vaak voor dat dezelfde verbeteringen vaak moeten worden herhaald voor iedere voorkomen van een naam. Dit is erg irritant maar er is geen andere mogelijkheid om dit sneller te doen. Het kan helpen om een vaak voorkomende verbetering te kopiëren en waar nodig te plakken.
**Let op**: soms is het onduidelijk waar een naam naar verwijst en soms verwijst een naam niet eenduidig naar iets (bijvoorbeeld "VIP's" of "een Hells Angel"). In dat geval moet in het chain-veld 'NONE' worden gezet!

Het **Type** geeft aan wat voor soort NE het is. Er zijn maar een beperkt aantal mogelijkheden:

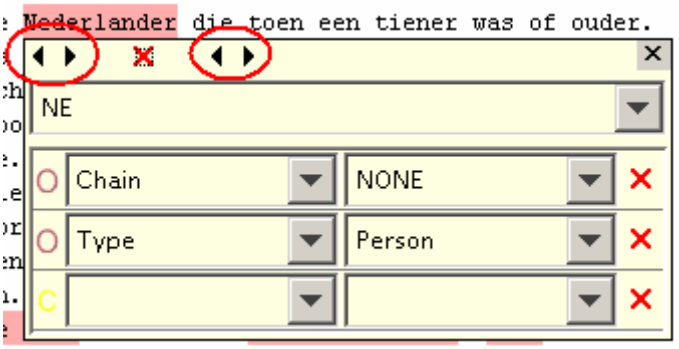
- *Person*: een enkel persoon, of een groep personen. Bijvoorbeeld 'Balkenende' en 'Nederlands voetbal elftal' hebben het type 'Person'.
- *Location*: locaties zoals steden, landen en geografische gebieden. Over het algemeen horen gebouwen NIET tot deze categorie (meestal horen die bij 'other').
- *Organization*: Organisatie of groep van organisaties met een formeel geassocieerde naam zijn van dit type. Organisaties kunnen zijn: overheid instanties (ministerie van volksgezondheid), bedrijven (KPN), sport groepen (PSV) en andere formeel georganiseerde groepen (carnavals vereniging 'de gekken').
- *Event*: iets dat tijdelijk en of periodiek optreed (een WK, verkiezingen, verjaardagen etc). Event wordt niet beschreven in de hieronder ACE handleiding, maar meestal is het wel duidelijk als het over een event gaat.
- *Other*: deze categorie is van toepassing als de bovenstaande categorieën niet van toepassing zijn op een NE.

Deze categorieën spreken over het algemeen voor zich. Als niks anders van toepassing is dient 'Other' te worden gekozen. Soms is het moeilijk om uit een type te kiezen, dit hangt vaak van de context af. In geval van twijfel moet de 'beste' worden gekozen. Wees hier wel consequent in! Specifiekere informatie over de types is te vinden in: http://projects.ldc.upenn.edu/ace/docs/English-Entities-Guidelines_v5.6.1.doc (let op, niet alle types beschreven in dit document worden gebruikt).
Voor alle NE's staat het type standaard op 'Person'. Alle NE's die geen persoon zijn moeten dus veranderd worden.

Het kan ook voorkomen dat de NE verkeerd is herkent. Er zijn verschillende soorten fouten:

- Het geselecteerde woord is in zijn geheel geen NE (vaak het begin van een zin). Verwijder simpelweg de annotatie zoals hierboven beschreven.

- Er is meer dan de daadwerkelijke NE geselecteerd (“van Holleeder”). Verwijder dan de foute annotatie en maak een nieuwe correcte. Of verander de tekst-selectie die bij de annotatie hoort door gebruik te maken van de kleine zwarte pijltjes bovenin het pop-upje. Zie figuur hieronder.
- Er is minder dan de daadwerkelijke NE geselecteerd ( Willem “Holleeder”). Verwijder dan de foute annotatie en maak een nieuwe correcte. Of verander de tekst-selectie die bij de annotatie hoort door gebruik te maken van de kleine zwarte pijltjes bovenin het pop-upje. Zie figuur hieronder.
- Een NE is in zijn geheel niet herkend. Maak dan een nieuwe correcte NE annotatie zoals eerder beschreven.



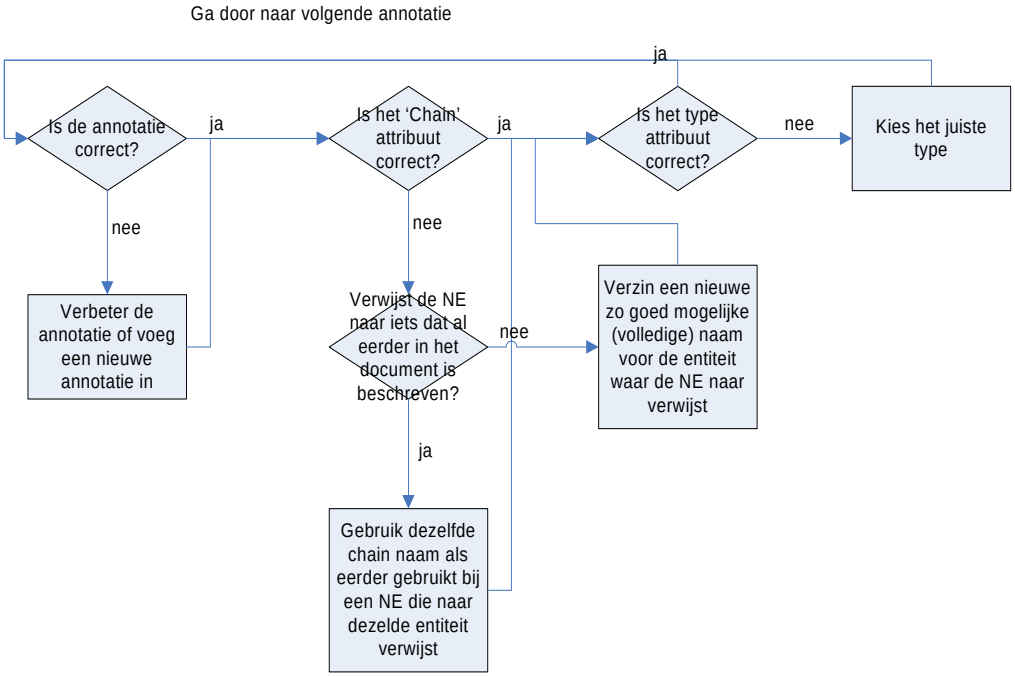

**Handig stappen plan**
Het makkelijkste is om gewoon het document van begin tot eind door te nemen en dan de annotaties te bekijken en zo nodig te verbeteren. Het is erg aantrekkelijk om in zijn geheel niet te lezen maar gewoon van annotatie naar annotatie te ‘springen’. Hierbij is echter het gevaar dat NE’s die het programma heeft gemist een tweede keer worden overgeslagen. Ook kan het dan lastig zijn om een goede ‘chain’ naam te verzinnen voor een nieuwe NE (omdat de context verder niet bekend is).

Als het programma een NE heeft overgeslagen dient deze dus handmatig geannoteerd te worden. Voor de meeste NE’s zijn gemakkelijk te herkennen en is duidelijk wat er precies bij hoort. Maar dit is niet altijd het geval. Selecteer dan de minimale tekst die zo precies mogelijk een entiteit in de wereld aanduidt (liefst zonder het gebruik van bijvoeglijke naamwoorden).

Ook komt het vaak voor dat een bijvoeglijk naamwoord een NE is maar dan is het onduidelijk waar het eigenlijk naar verwijst. Een voorbeeld hiervan is “de Amsterdamse politie” waarin alleen “Amsterdamse” als NE is aangegeven. Het geheel verwijst specifiek naar een bepaald deel van een organisatie (niet zomaar de politie maar de Amsterdamse politie). Het geheel dient dan ook als NE te worden geannoteerd. Helaas is het niet altijd duidelijk waar een dergelijke constructie naar verwijst of de verwijzing is niet eenduidig.

Aangezien de documenten aan elkaar gerelateerd zijn is de kans groot dat een aantal namen in verschillende documenten voor komen. Deze namen moeten dus allemaal dezelfde chain-waarde krijgen en daarom kan het makkelijk zijn om de gebruikte waarden in een tekst documentje op te slaan. Op die manier kan je gemakkelijk bekijken of je een naam al eerder hebt geannoteerd en met welke waarde.

Per annotatie kan de workflow, zoals in de figuur hieronder aangegeven, worden gebruikt. Dit is niet verplicht maar het is een makkelijke 'handleiding' om niets te vergeten tijdens het annoteren. De 'is de annotatie correct' vraag is het start punt.




Annoteer nooit te lang en neem pauzes! Als je lang achterelkaar dit (saaie) werk doet ga je snel dingen over het hoofd zien (en het is ook niet goed voor je ogen etc om steeds naar het scherm te turen).

**Het uiteindelijke resultaat**

Als alle documenten zijn geannoteerd moeten ze naar mij worden teruggestuurd. Zip of rar de directory met de geannoteerde documenten (en eventuele tekst bestanden met chain waarden etc). Stuur dit dan naar mij op (corne.versloot@tno.nl) zodat ik er verder mee kan werken.

**Tips & Tricks**

Annoteren in GATE gaat relatief gemakkelijk maar het programma heeft wel een paar rare 'eigenschappen'.

- Bij het veranderen van een annotatie gebeurt het soms dat het pop-upje ineens verdwijnt. Meestal komt dit doordat de muis dan op een andere annotatie terecht is gekomen en dan daarvan het pop-upje activeert. Als je in een dergelijk geval net iets aan het typen bent dan is de kans groot dat je in het document zit te typen. Dit is absoluut niet de bedoeling, er mag geen extra tekst in het document worden ingevoerd!
- GATE heeft geen CTRL+Z (terugdraai) functie. Dus als er iets fout gaat moet het handmatig worden teruggedraaid!
- Gebruik nooit 'Close' op een document omdat dit het betreffende document uit het corpus verwijdert. Gebruik i.p.v. 'Close' 'Hide this view'!
- Als een document door de hide operatie uit het middelste scherm wordt verwijderd word in het linker frame in de lijst met documenten dit document geselecteerd. Dit klinkt logisch, maar als je op dat moment een ander document (bijvoorbeeld die je wilde gaan annoteren) had geselecteerd veranderd dus die selectie. Als je dan het geselecteerde document opent om aan te beginnen blijkt die dus al gedaan te zijn!

- Annoteer nooit te lang en neem pauzes! Als je lang achterelkaar dit (saaie) werk doet ga je snel dingen over het hoofd zien (en het is ook niet goed voor de ogen etc om steeds naar het scherm te turen).
- Het type staat altijd op 'person' en in 75% van de gevallen is dit correct. Als in eerste instantie veel personen voorkomen en dan ineens iets anders is het gevaar dat wordt vergeten om het type goed te zetten!
- Veel documenten bevatten rare tekens (vraagtekens etc). Dit komt verschil van tekst formaat. Ook al is dit natuurlijk fout, het is niet de bedoeling om aan te passen.

**Voorbeelden**
Hieronder volgen een aantal voorbeelden van 'probleem gevallen' en de oplossingen daarvan. Dit zijn echt voorkomende voorbeelden uit het corpus, de NE's zijn dikgedrukt.

"…de eerste groepsduels van **Oranje** bij het **WK-voetbal** in **Duitsland** woekerprijzen…"
Oranje: chain(nederlands voetbal elftal), type (Person). (groep van mensen, kan ook Organization zijn maar meestal wordt echt het 'team' met deze term aangeduid)
WK-voetbal: chain (Wereld kampioenschap voetbal 2004), type (Event) (er zijn verschillende WK's in verschillende jaren, dus simpelweg WK is niet genoeg)
Duitsland: chain(Duitsland) type(Location)

"..,de vijfde plaats van het **WK** in **Rusland** stuit…"
WK: chain( wereld kampioenschap vrouwen handbal 2004)(!!) chain(Event)
Rusland: chain(Rusland), type(location)
Het gaat hier over Nederlandse vrouwen handbal team en dus niet over het normale WK (dat normaliter voor voetbal wordt gebruikt)

"…benijden in de **Kuip**, zoals hij…"
Kuip: chain(voetbalstadion de Kuip), type (other) (het is geen locatie)

"Maar in het **Wilhelmus** staat niet **Diets**, er staat…"
Wilhelmus: chain(Nederlands volkslied), type(Other)
Diets: chain(NONE) (het verwijst niet direct naar iets), type(NONE)

"… dat de **Duitsers** hun…"
Duitsers: chain(NONE), type(Person)
(de term verwijst niet duidelijk naar iets of iemand (afgezien van een heel volk, maar het gaat in ieder geval wel over personen))

"…het **WK van Beckenbauer** weer…"
WK van Beckenbauer: chain(NONE), Type(Event).
Dit is een zeer onduidelijke NE dat verwijst naar een bepaald concept. Hiervan is niet goed op te schrijven wat het is of van welk type!

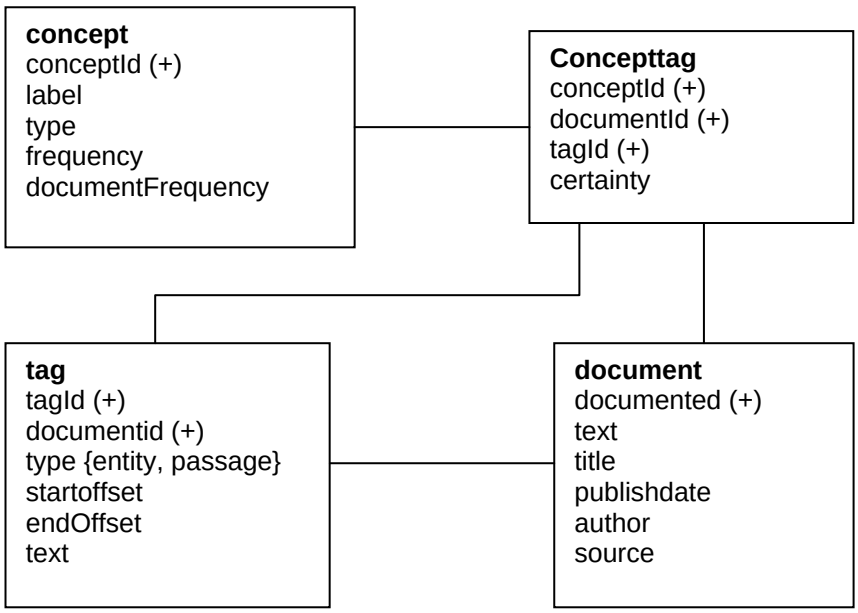
"…Onder andere in de **Volkskrant** van 11…´
Volkskrant: chain(Volkskrant), type(Other).
Het gaat over de krant zelf niet over de organisatie.

"… journalist van het **AD** meldde en forse stijging…"
AD chain(algemeen dagblad), type(Organization)

In het eerder ook al genoemde: http://projects.ldc.upenn.edu/ace/docs/English-Entities-Guidelines_v5.6.1.doc zijn meer voorbeelden te vinden (voor het engels, maar dat maakt niet echt uit).

# Appendix B

The new Novalink database structure. Every part in the text can be represented by an entry in the tag table. In case of this research these parts are always names that belong to exactly one concept (i.e. named entity). Using the concepttag table the tags and their documents can be linked to one concept.



# Appendix C

This appendix shortly describes the different possibilities tried to improve SVM's as indicated in chapter 8.5.3.

Optimized gamma and cost parameters
Two parameters can be set before training an SVM with RBF kernel function: cost (C) and gamma (g). The cost parameter is the tolerance to error, gamma is the width of the Gaussian for the RBF kernel. An automated grid search (by a lib-SVM python script) was used to optimize the gamma and Cost parameters. Unfortunately SVM's trained using optimized g and C values did not perform better than the basic SVM.

One class SVM
Because the data is so unbalanced it is hard for an SVM to learn. One option is to train the SVM on only one class so it can lean if new samples belong to the class or not. In this case there are only two classes that can be used for this process.

Using LibSVM two one-class models were trained (one for the negative and one for the positive samples). Both SVM's were trained three times on 10000 samples and evaluated on the same evaluation set as used before. The average results are shown below:

| | |
|---|---|
| **Accuracy** | 86.83% |
| **Precision** | 0.0036 |
| **Recall** | 0.47 |
| **F-measure** | 0.007 |

Postivie one-class results

| | |
|---|---|
| **Accuracy** | 50.29% |
| **Precision** | 0.0026 |
| **Recall** | 0.13 |
| **F-measure** | 0.0052 |

Negative one-class results

The results are clearly less good than the normal SVM. It is obvious that the two classes are not homogeneous enough to be learned in this way. Strangely enough the negative samples are very hard to learn.

Optimize SVM for F-measure
Since SVM's optimize their model on the accuracy of the training data it is not likely that a better model will be learned (since the accuracy already is 99.89%). LIBSVM has the option to optimize the SVM not on accuracy but on F-measure. Unfortunately the package was not yet fully developed and optimization on F-measure could only be done using a linear kernel and not using a RBF Kernel. The F-measure optimization using a linear kernel resulted in a model with an accuracy of 83.56%, a lot lower than the 99.89% accuracy of the standard (RBF-kernel) SVM.

# Appendix D

### Novalink Evaluatie

In deze evaluatie is het de bedoeling om een groot aantal afbeeldingen te becijferen. Het doel van de afbeeldingen is om een correct en duidelijk beeld te geven van de omgeving van iets of iemand. De omgeving bestaat uit namen van personen, plaatsen en organisaties die een relatie hebben met de zoekterm. In dit onderzoekje worden steeds 2 verschillende overzichten van een bekende naam gegeven. Het is de bedoeling dat u beide overzichten gaat becijferen met een getal tussen de 1 en de 10. Hierbij moet u niet letten op de verschillende kleuren in de afbeeldingen. Het antwoord op de volgende vraag het belangrijkste criterium:

*Welke graaf geeft het beste overzicht als je niets weet van de gebruikte zoekterm*

Hierbij moet u zich dus voorstellen dat u weinig tot niets weet van de gezochte naam, welk overzicht biedt dan de meeste en/of beste informatie. Er een aantal dingen waar u op kunt letten:

- Zijn de gevonden relaties correct?
- Is het geheel informatief?
- Zijn de relaties en namen duidelijk?

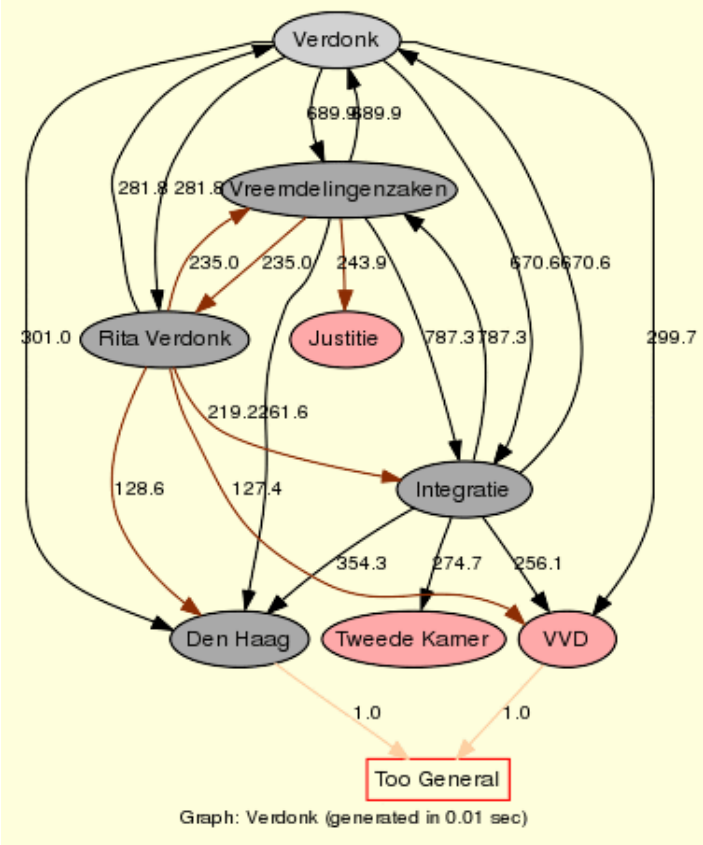
Het is de bedoeling dat u een korte argumentatie geeft voor de becijfering. Voorbeelden kunnen zijn dat er:

- Goede en nuttige relaties in het overzicht zitten
- Een belangrijke relaties mist
- Relaties fout zijn
- Relaties niet echt informatief zijn
- Namen onduidelijk zijn
- Etc.

Bij twijfel of een naam wel in de afbeelding thuis hoort is het toegestaan om even te ‘googlen’ om uit te vinden wat die naam te betekenen heeft en of er daadwerkelijk een relatie bestaat met de zoekterm.

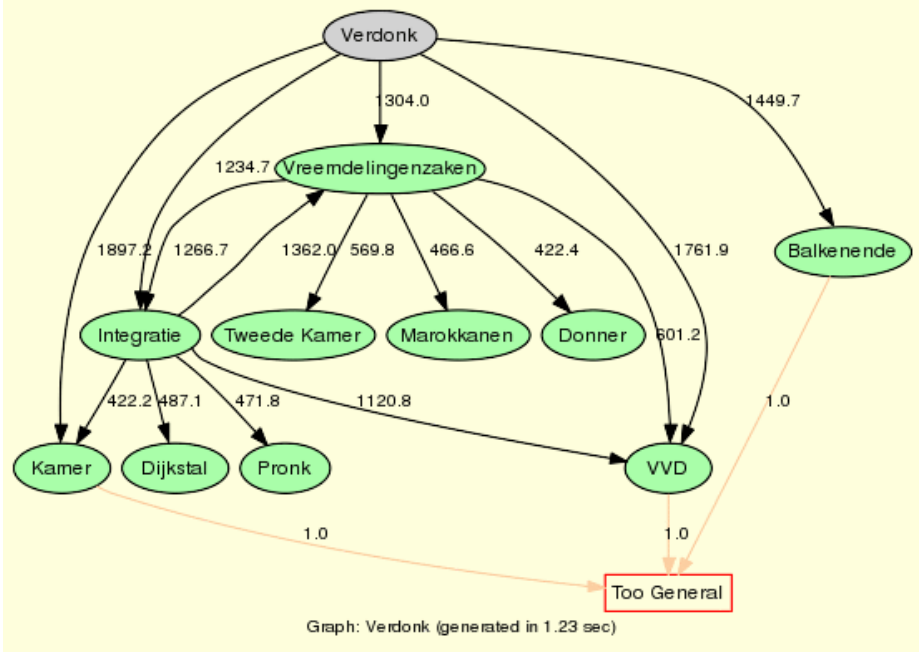
Veel succes en bedankt!

Zoekterm: Verdonk



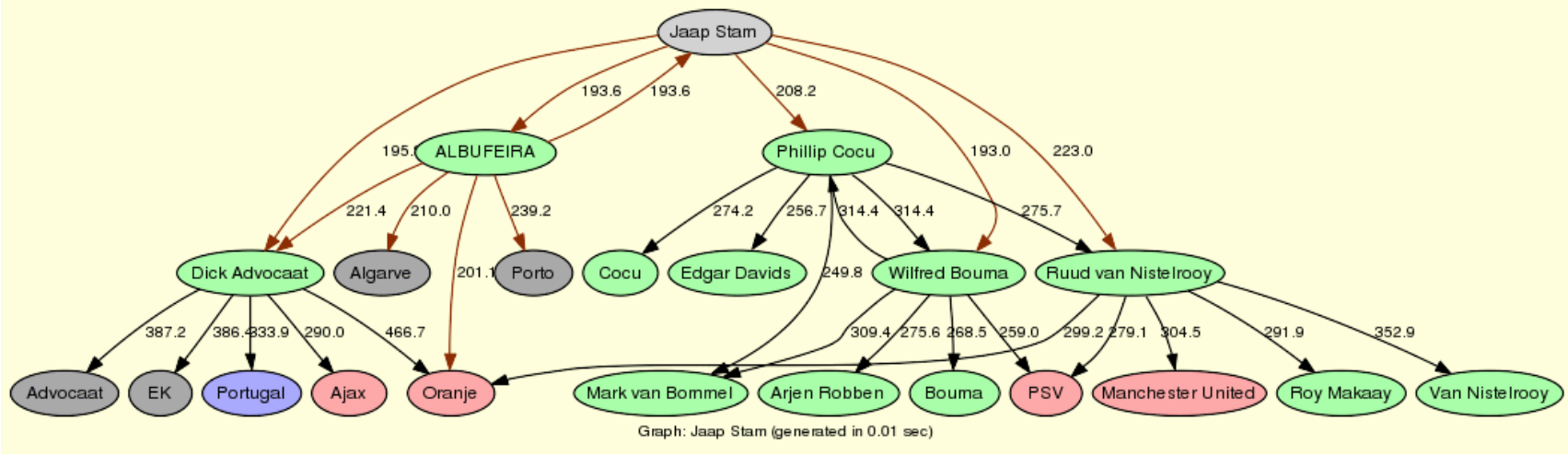

Average rating: 6,89

Zoekterm: Verdonk



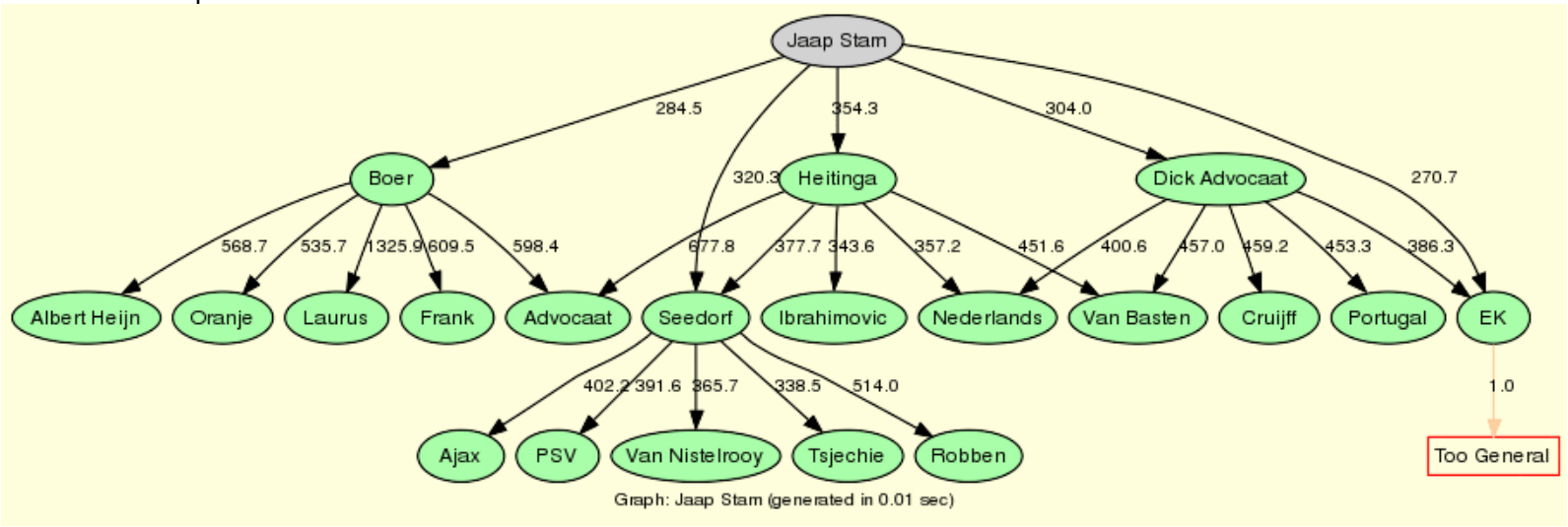

Average rating: 7,33

Zoekterm: Jaap Stam



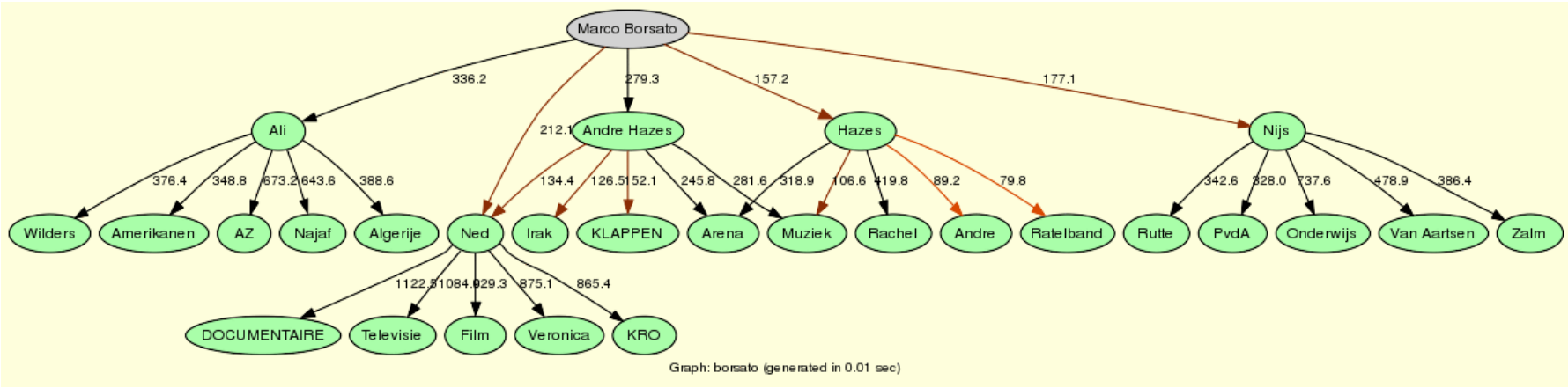

Cijfer: 6.71

Zoekterm: Jaap Stam



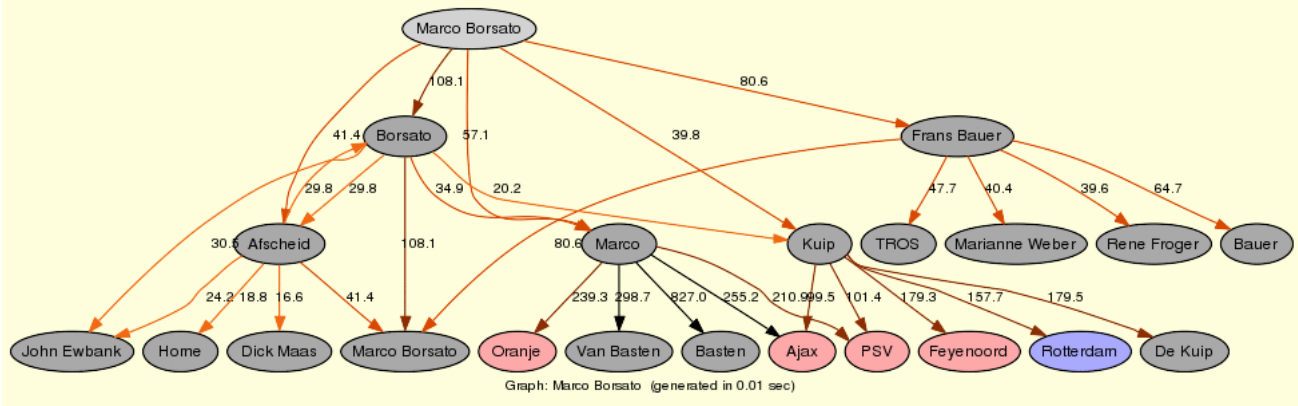

Cijfer: 6.57

Zoekterm: Marco Borsato



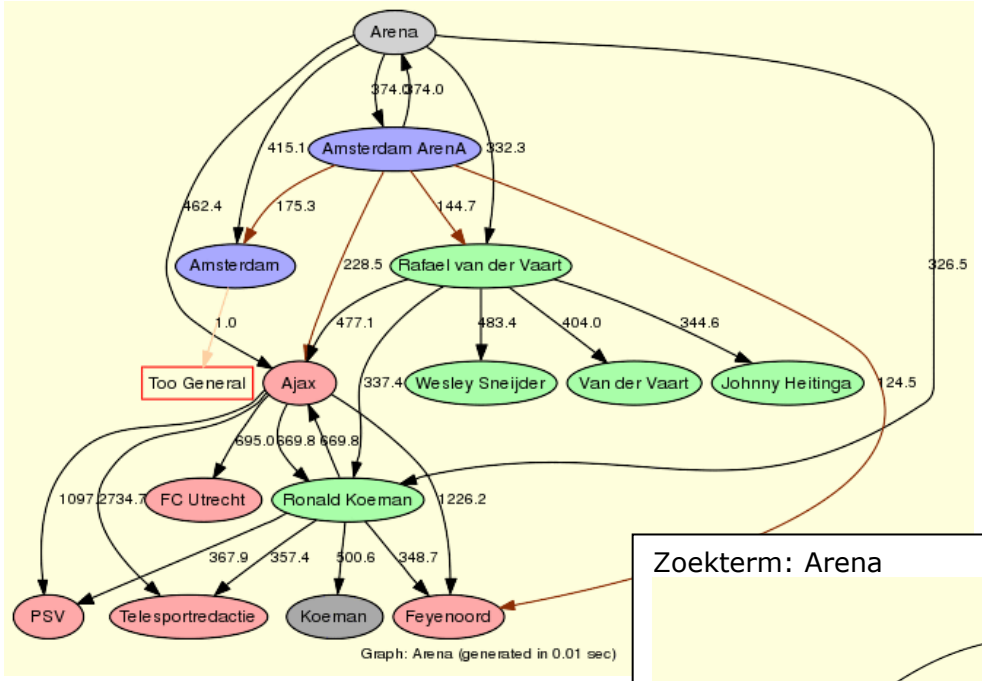

Average rating: 6,22

Zoekterm: Marco Borsato



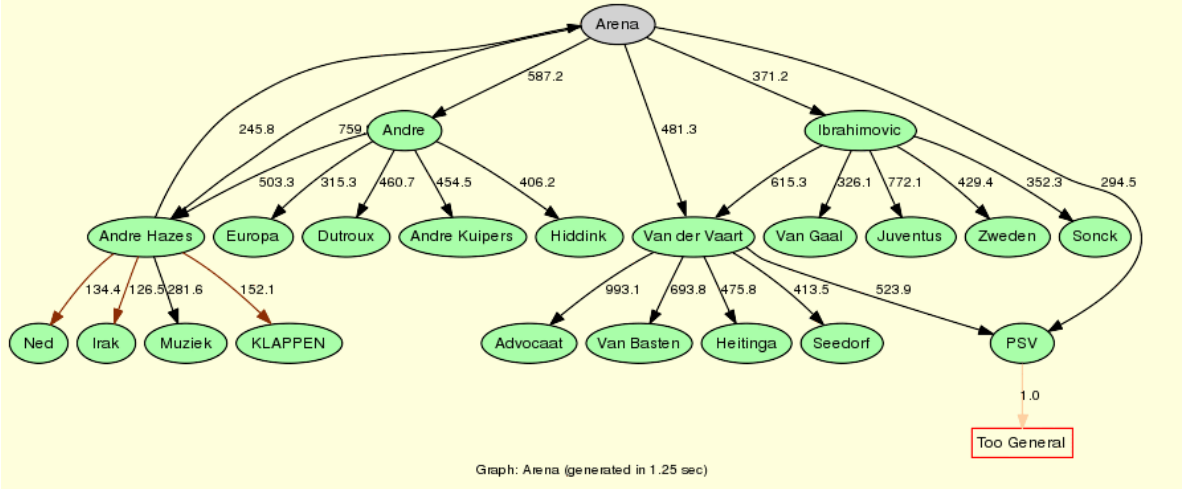

Average rating: 6

Zoekterm: Arena



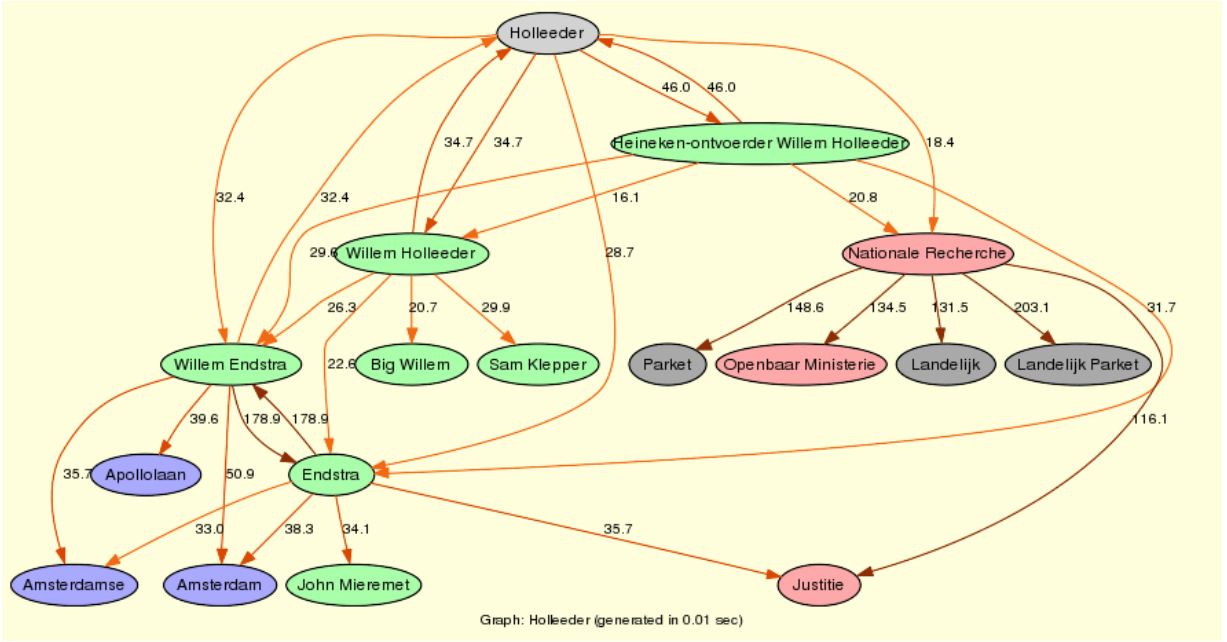

Average rating: 6,33

Zoekterm: Arena

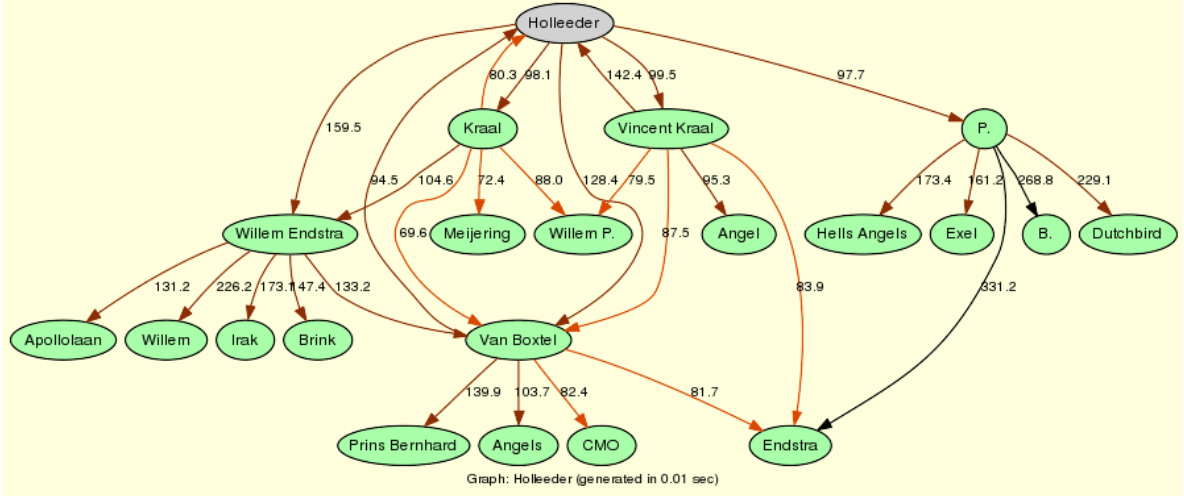



Average rating: 6

Zoekterm: Holleeder

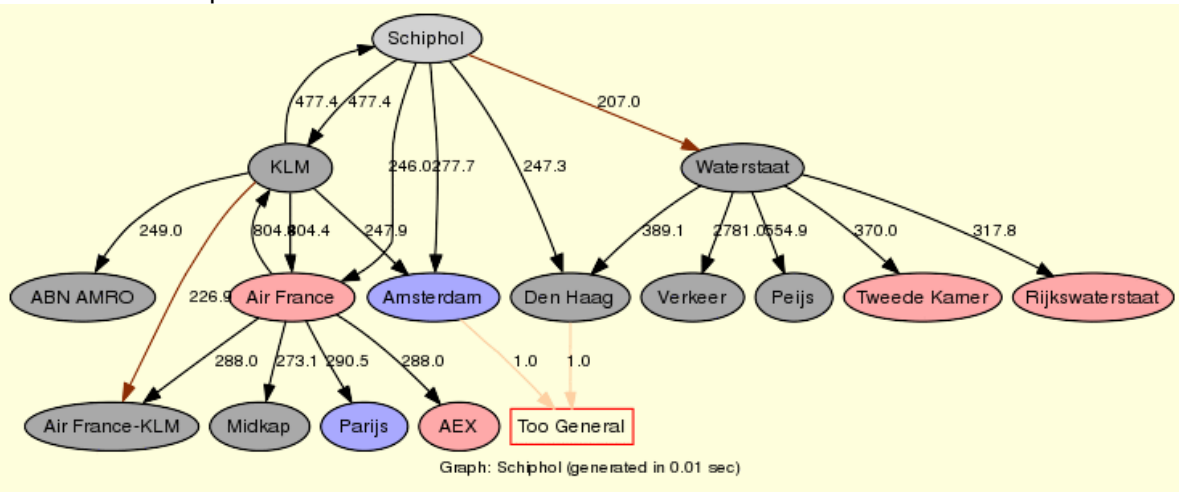



Average rating: 7



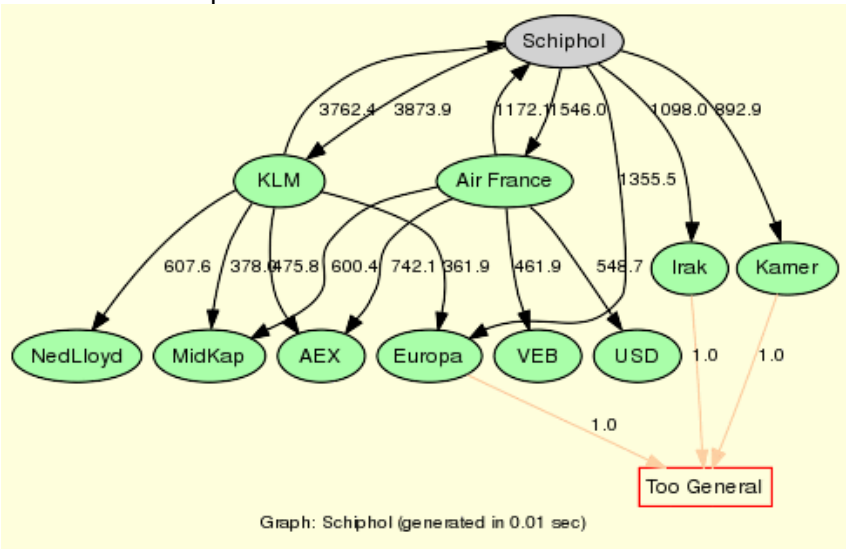

Average rating: 6,78

Zoekterm: Schiphol



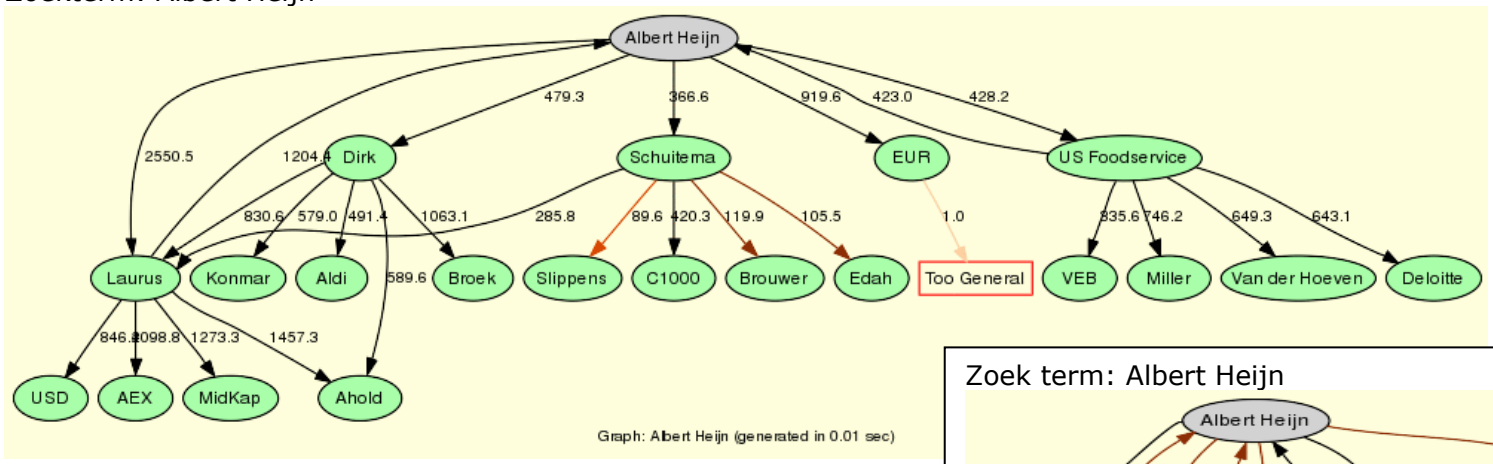

Average rating: 6,78

Zoekterm: Schiphol



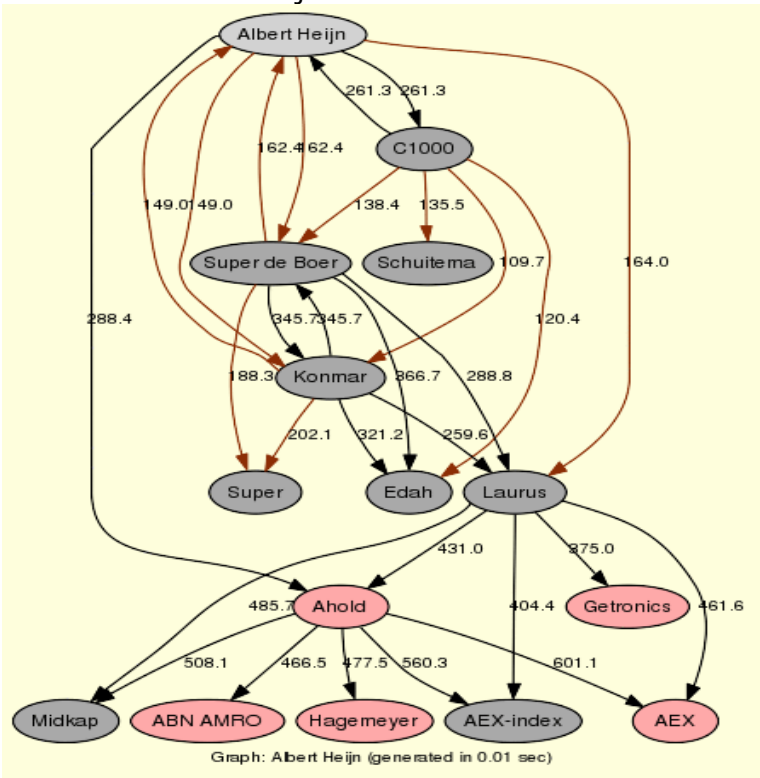

Average rating: 6,33

Zoekterm: Albert Heijn



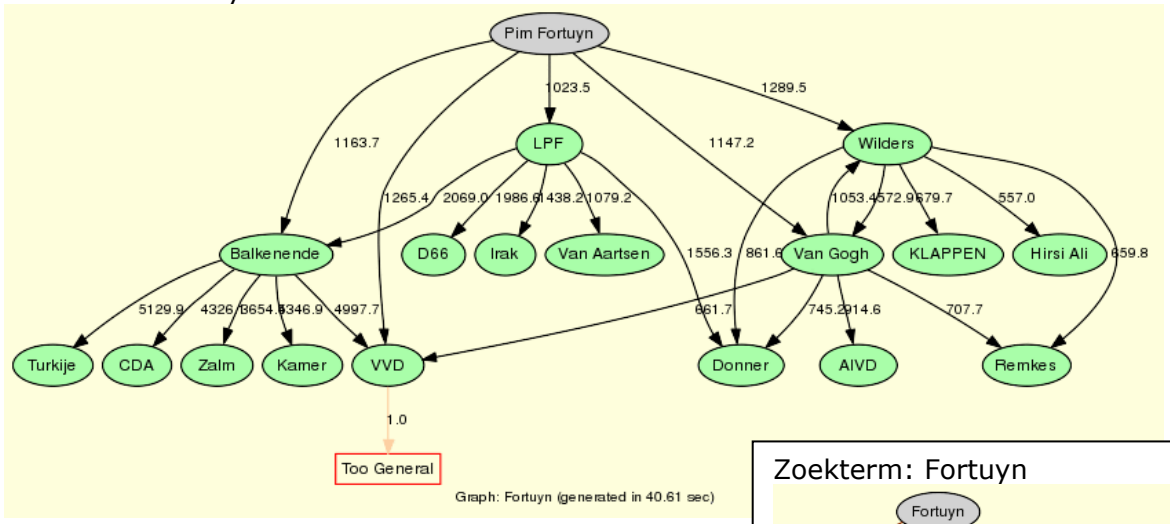

Average rating: 7

Zoek term: Albert Heijn



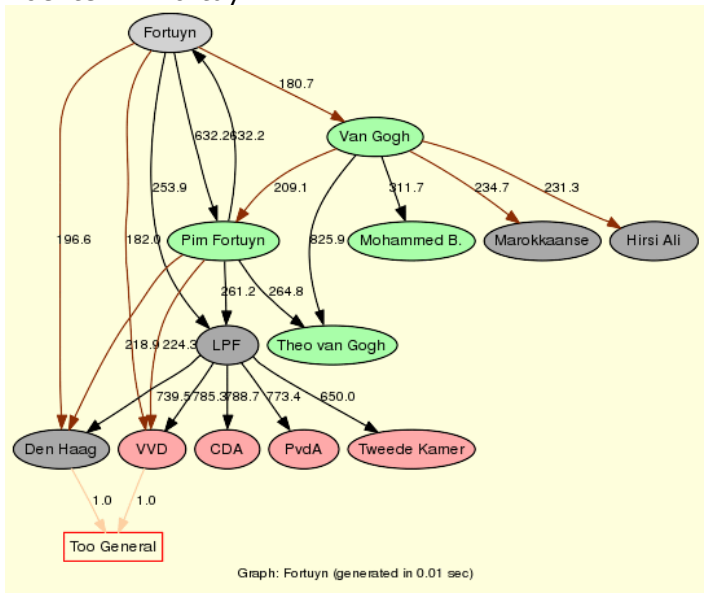

Average rating: 6,56

Zoekterm: Fortuyn



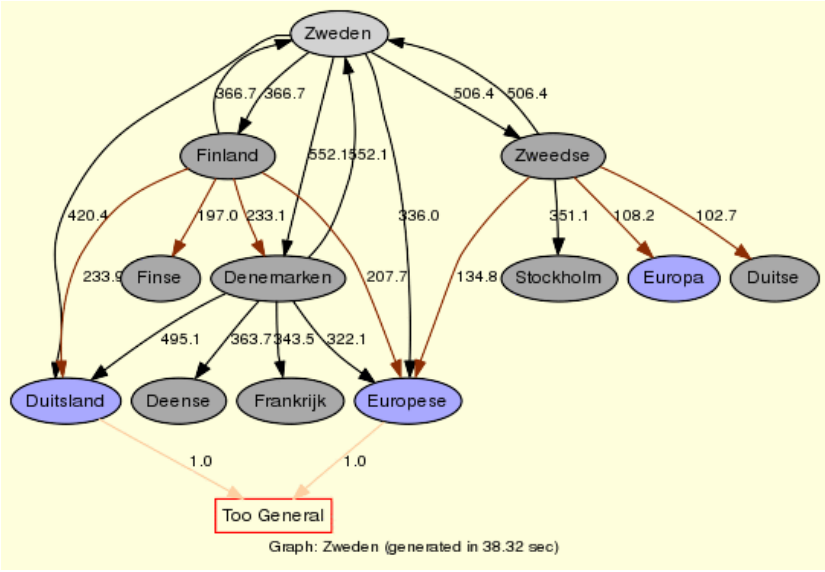

Average rating: 7,17

Zoekterm: Fortuyn



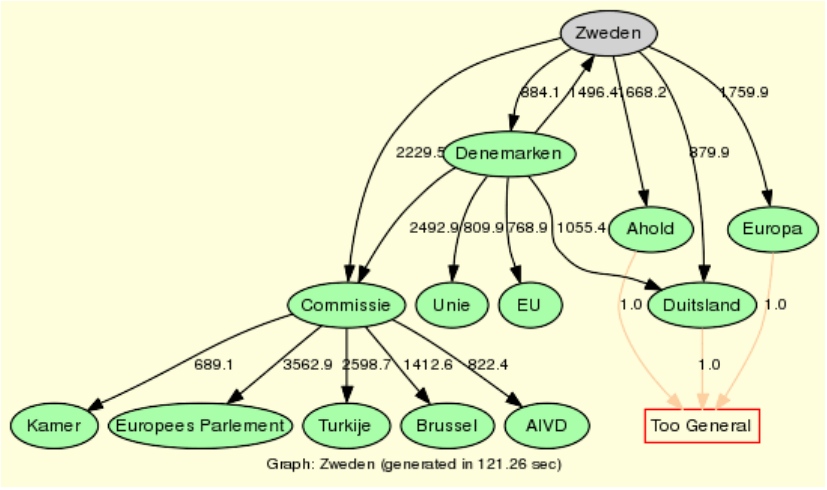

Average rating: 6,67

Zoekterm: Zweden



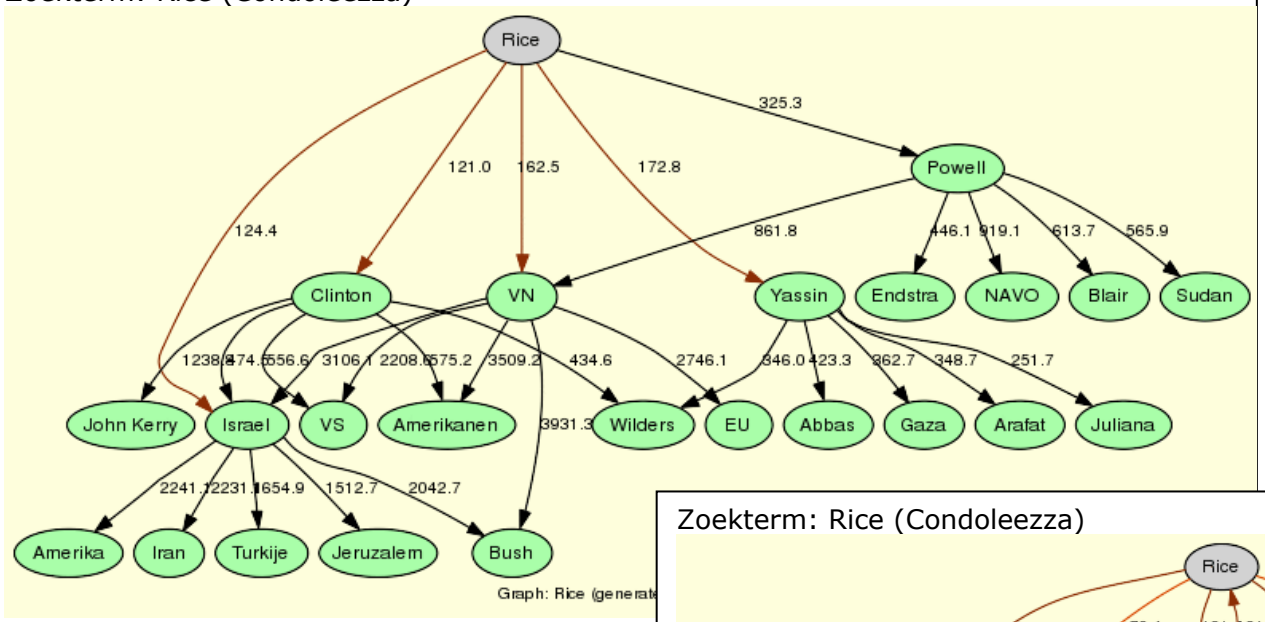

Average rating: 6,43

Zoekterm: Zweden



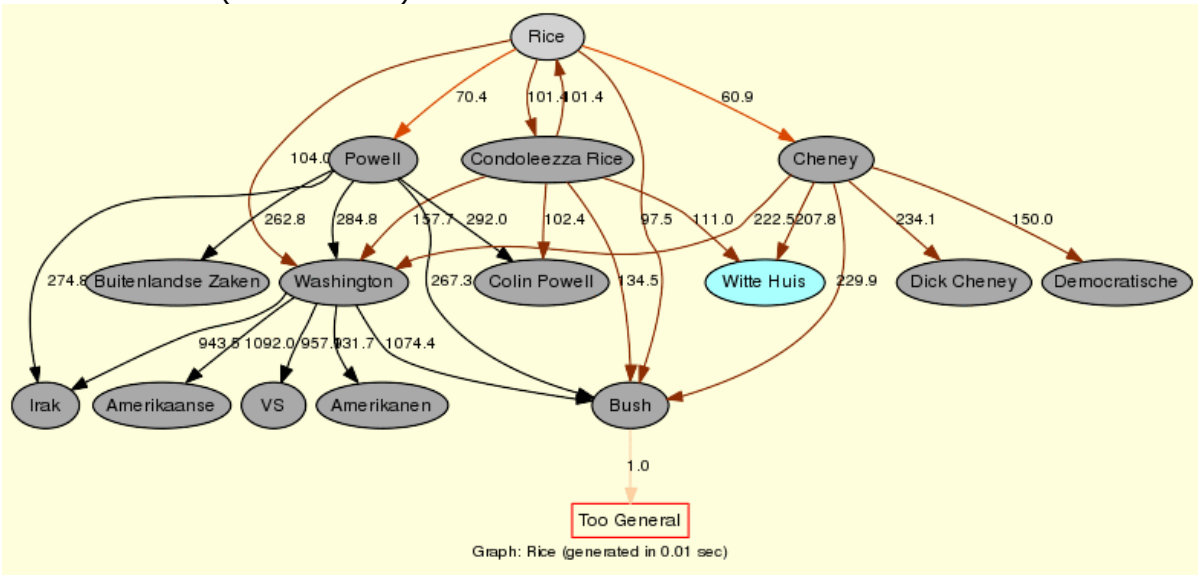

Average rating: 6,43

Zoekterm: Rice (Condoleezza)



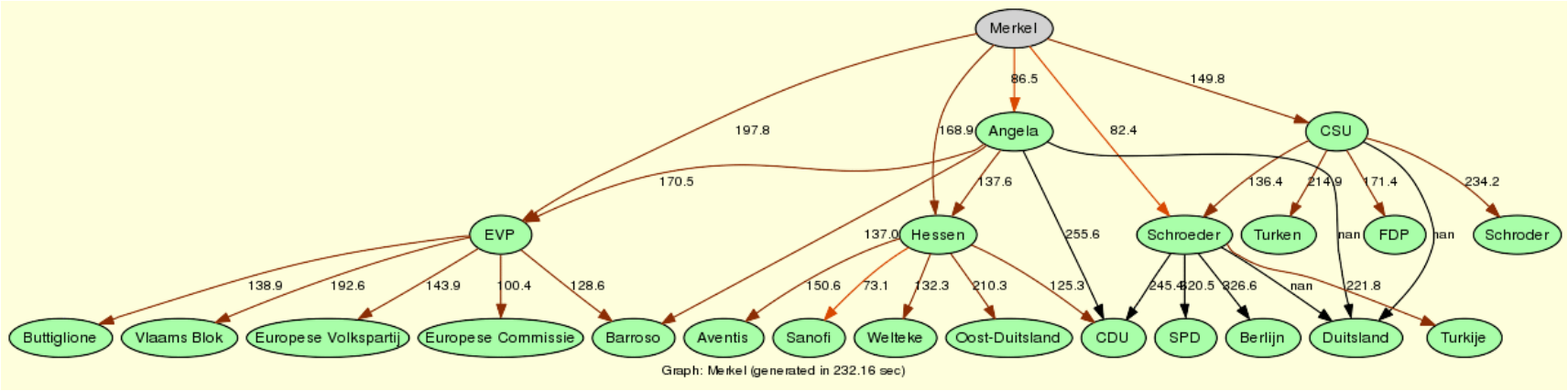

Average rating: 7

Zoekterm: Rice (Condoleezza)



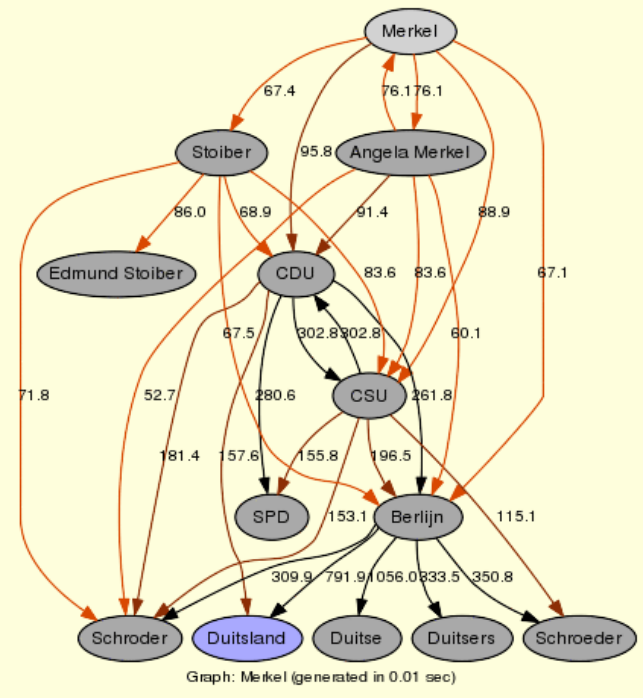

Average rating: 7,14

Zoekterm: Merkel



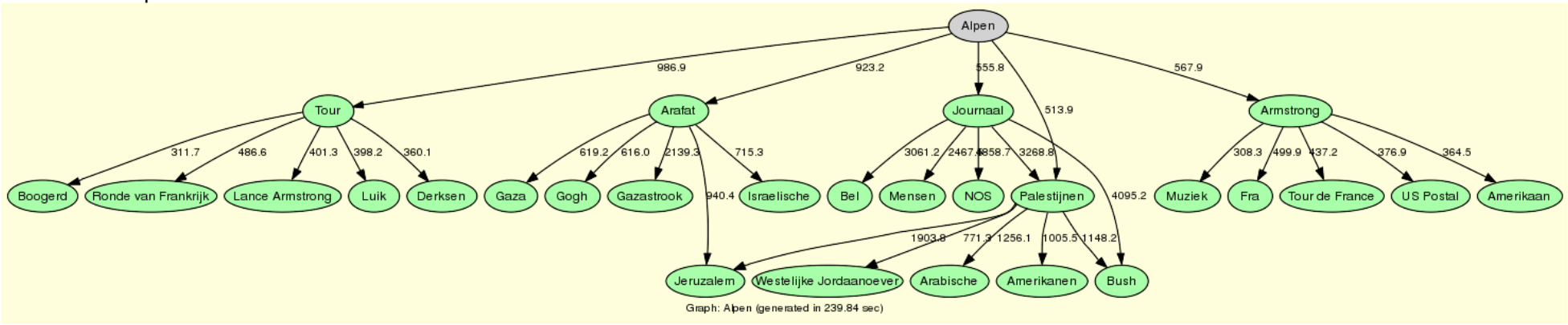

Average rating: 7,14

Zoekterm: Merkel



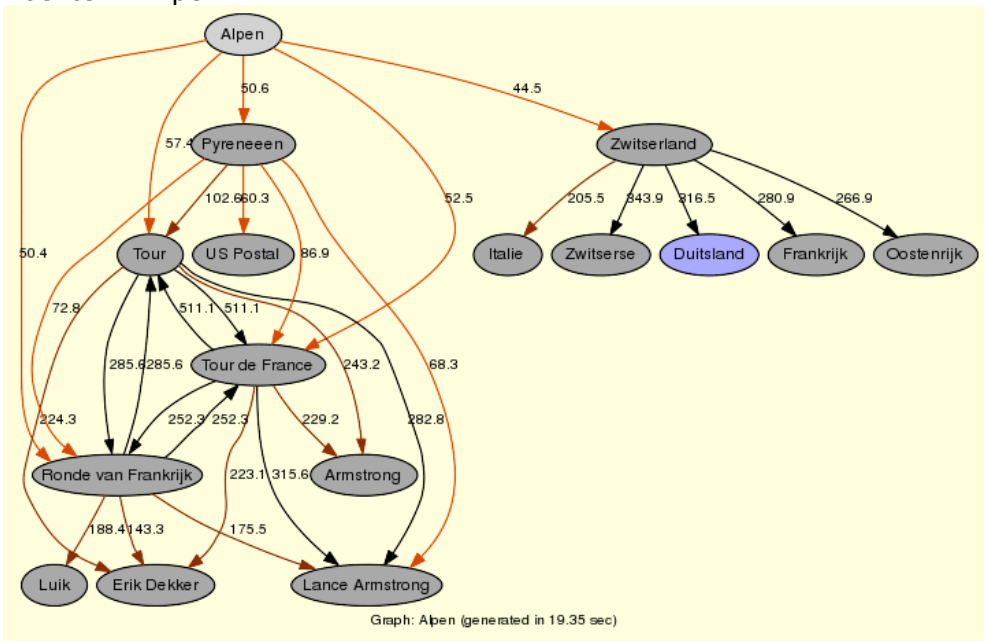

Average rating: 7

Zoekterm: Alpen



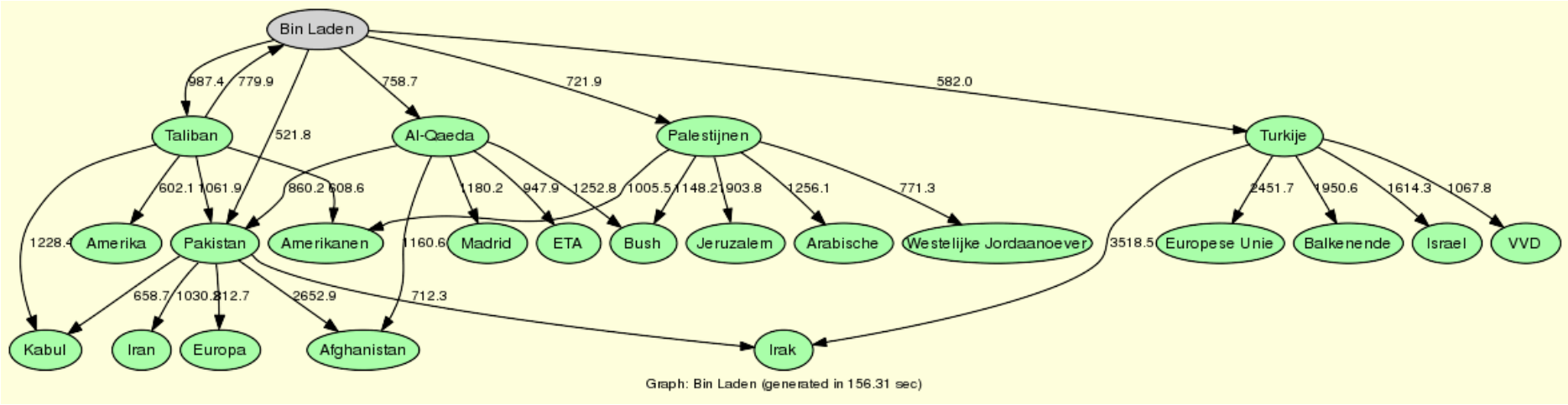

Average rating: 5,57

Zoekterm:Alpen



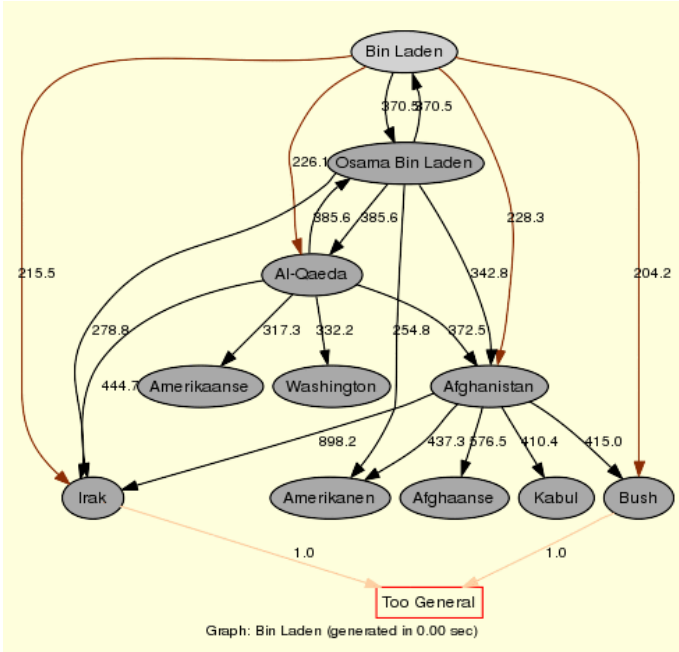

Average rating: 7

Zoekterm: Bin Laden



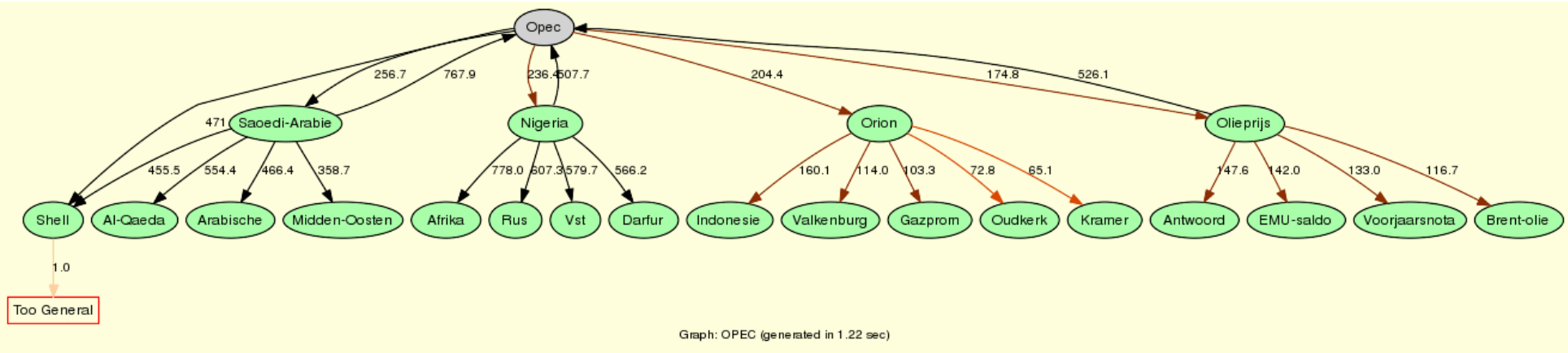

Average rating: 7,86

Zoekterm: Bin Laden



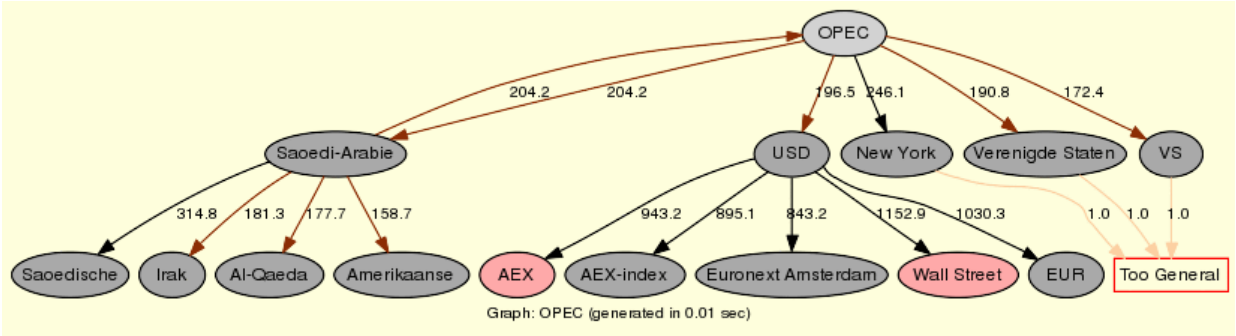

Average rating: 6,14

Zoekterm: OPEC



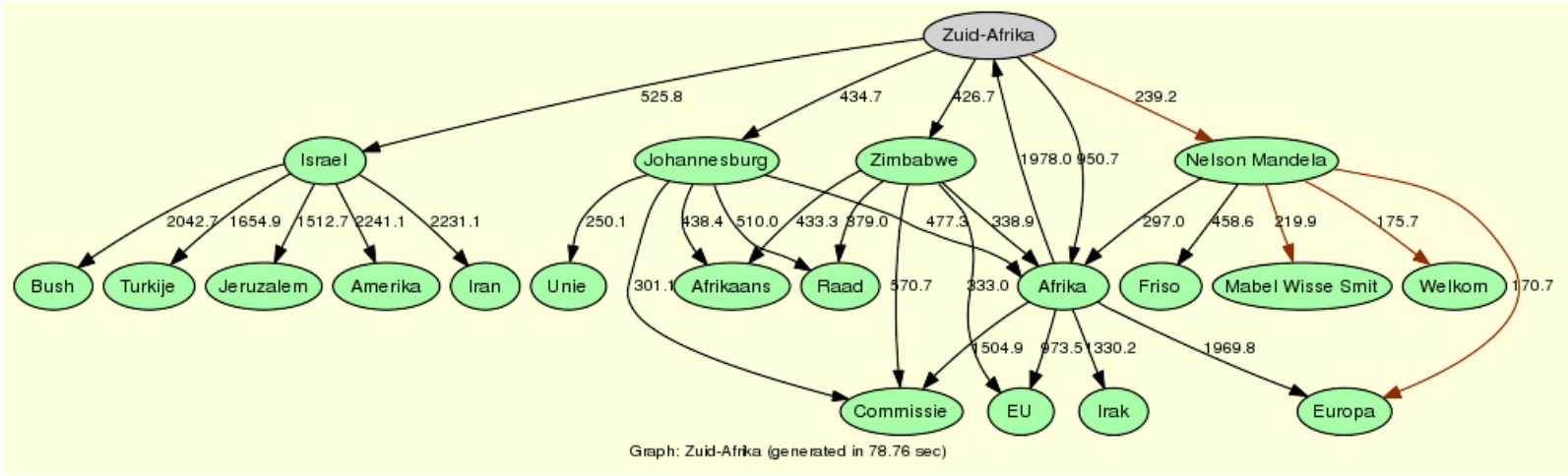

Average rating: 7

Zoekterm: OPEC



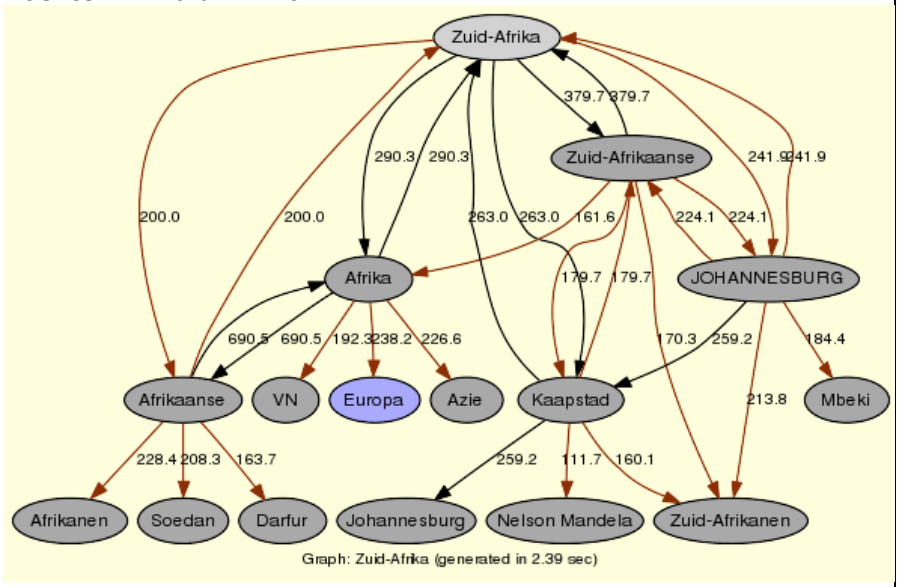

Average rating: 6,43

Zoekterm: Zuid-Afrika




Average rating: 6,71

Zoekterm: Zuid-Afrika




Average rating: 7,14