UNIVERSITY OF GOTHENBURG

DEPARTMENT OF PHILOSOPHY, LINGUISTICS AND THEORY OF SCIENCE

ENTITY RELATION EXTRACTION

Exploring the Use of Coreference Resolution in a Distant Supervision Approach to Automated Relation Extraction

Tessa Koelewijn

Master's Thesis:	30 credits
Programme:	Master's Programme in Language Technology
Level:	Advanced level
Semester and year:	Spring, 2016
Supervisor	Richard Johansson
Examiner	Lars Borin
Report number	
Keywords	NLP, relation extraction, machine learning, coreference resolution

Abstract

This Master's thesis describes the effect of coreference resolution on a distant supervision approach to automated relation extraction. Coreference resolution is used as an addition to an existing relation extraction method, described in Mintz (2009). The thesis gives a detailed analysis of a reimplementation of this existing method, and provides an insight in how coreference information influences the performance. In this distant supervision approach to relation extraction, the database Freebase is used for the annotation of relations in Wikipedia text. A classifier is then trained to learn these relations between entities in the text. The main advantage of this approach is that the data is automatically annotated. This prevents the expenses of manual annotation, and makes it possible to train a classifier on a very large dataset. Using coreference information is a way of increasing the amount of data available and the expectation is that this will improve the result of the relation extraction system. An automatic evaluation method and a manual analysis of the performance are described, providing a detailed insight in the system's behaviour. The evaluation shows that including coreference information does not improve the precision and recall of the system. However, it does enable the relation extraction system to find more relation instances, which shows that is does have an advantage and a lot of potential in future research.

Acknowledgements

I would like to thank my supervisors, Richard Johansson and Jonatan Bengtsson, who have not only given me the guidance and help I needed to complete this thesis, but also have contributed with interesting discussions that always have revived my enthusiasm for the topic. Special thanks also to the company Findwise, which has helped me with the necessary materials, has provided a very friendly and comfortable place for me to work and has brought me in contact with many interesting people. Last but not least, I would like to thank my closest friends, who have given me all the moral support and encouragement I could wish for.

Contents

1 Introduction

This thesis will describe a method for automated relation extraction. The goal is to give a detailed analysis of the distant supervision approach described by Mintz (2009) and to explore the usage of information obtained by coreference resolution as an addition to this approach.
The problem of automated entity relation extraction can be defined in the following way:

(1) how are entities in a text related to each other and how can those relations be automatically extracted from unstructured texts.

In order to be able to develop a system that can extract relations from text, it is important to define the problem in a more specific way. First of all, it needs to be clear what is considered a relation and what is considered an entity.

As a very general definition, a relation can be anything that ties two or more entities together, but in this thesis and many other investigations it is restricted to relations between two entities. A relation provides information about the participating entities that is not available from the entities themselves, and therefore goes beyond the information that can be obtained by, for example, a named entity recogniser. By using a named entity tagger, it is possible to see if two entities appear in a text or sentence together. This could indicate that those entities are somehow related, but by just looking at the co-occurrence, it is impossible to say how they might be related. Say that these entities are a person and a location, a relation extractor would be able to determine whether the person is born in this location, or living in this location, or is part of the government of the location. It would be ideal if a relation extractor would be able to detect relations from any two entities that appear in the same document, but the relation extractor described in this thesis is limited to entities that appear in the same sentence.

An important requirement for a relation extraction system is to determine what kind of relations can be extracted. For unsupervised methods, this is often an open set, so any type of relation can be found. Any sentence predicate could be a relation in this case, and the result would be a very flexible system with a great variety of different relations. A disadvantage is that there is no control over the informativity of the relations it finds. This means that these type of systems often include very general relations, such as 'is a'. Even though these relations can be useful, it can be an advantage to determine in advance which relations are relevant for the situation.

In (distant)supervised methods, a closed set of relations is determined by either human annotation (fully supervised), or by some database (distant supervision). The main advantage of using a predetermined set of relations, is that the relation extraction system can be restricted to only find information that is useful for the situation. In the relation extraction system developed in this thesis, the set of relations is determined by Freebase (Bollacker et al., 2008), a database that consists of many relations and examples of entity pairs that have these relations. The Freebase data used for the baseline implementation is manually filtered for the purpose of the approach described in Mintz (2009), so that redundant relations are removed or merged. In this thesis, two different sets of relations will be implemented and evaluated. One is identical to the set of relations used in Mintz, and in the other set the relations are re-filtered, so that only relations that connect persons, locations, organisations and entities that typically belong to the miscellaneous class are included. This re-filtered set is made to make sure the relations in the database match the types of entities that can be recognised by the named entity tagger.

In order for a relation extraction system to be able to perform the task of detecting relations, it must be clear what entities will be considered as possible candidates for participating in a relation. In some unsupervised methods for relation extraction, such as Banko (2007), many different elements will be considered an entity, such as proper nouns and noun phrases. However, in approaches that use some type of supervision, such as fully or distant supervised methods, typically a named entity tagger is used to determine the candidate entities. The one used in this thesis is the Stanford CoreNLP (Finkel, Grenager, & Manning, 2005) Named Entity Recognizer, which can detect a wide variety of different entities in text. However, only the following four categories will be considered: persons, organisations, locations and a miscellaneous class that typically contains names such as locations that are used as an adjective. Words such as 'English' in 'English language' will be tagged as miscellaneous.

The relation extraction system described in this thesis locates entities in a document by using a named entity recogniser and identifies whether there are relations between the entities and what those entities are. The relations that can be recognised by the system, are the relations that occur in the Freebase data. The output of the system are tuples containing the entity pairs and the relations they participate in. An example of such a tuple would be (Obama, people/person/place_of_birth, US), indicating that Obama is born in the US.

In addition to a detailed analysis of the relation extraction system as described in Mintz (2009), the contribution of this thesis is the usage of coreference resolution in this approach. This means that the system will not only take the literal mentions of entities into account, but also other words referring to those entities, such as pronouns. The expectation is that using coreference resolution increases the amount of information available and will increase the performance of the system.

1.1 Motivation

Existing studies have implemented various approaches to automated relation extraction, varying from supervised (Zhou et al., 2005; Kambhatla, 2004), unsupervised (Shinyama & Sekine, 2006) and distant supervised (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2012; Hoffmann et al., 2011) learning methods, sometimes combined with neural networks (Zeng et al., 2015).

The main issue with supervised approaches is that it requires a lot of manually annotated data, which is both time consuming and expensive. Also, the resulting systems are often very domain dependent.

Unsupervised methods perform fairly well, but the resulting relations are quite unpredictable and often not as informative as systems that use a predefined set of possible relations as a basis.

This thesis will describe a relation extraction system based on the distant supervision approach described in Mintz (2009). This machine learning method uses a database of facts to supervise a large amount of unlabelled data. It makes use of the relations in Freebase (Bollacker et al., 2008), a database containing a lot of relations and entities participating in those relations. For example, the relation people/person/place_of_birth contains many tuples of people and locations. The training and testing data will be annotated with the entity pairs and relations present in Freebase, removing the need for human annotations.

A reimplementation of the approach described in Mintz (2009) will serve as a baseline to which the additions made in this thesis will be compared. The most important assumption of the distant supervision approach used in the baseline is the following:

(2) when two related entities occur in the same sentence, the relation between them will be expressed in some way in that same sentence.

This assumption means that every sentence that contains two entities that are related according to Freebase, will be treated as an expression of this relation. This is an overgeneralisation, but it allows the data to be treated as annotated data. Every sentence that contains two related entities will be annotated for that relation. These annotations then form the data on which the classifier will be trained. This means that even though the distant supervision approach does not use manually annotated data, the learning algorithm has a lot in common with supervised approaches.

The advantage of this distant supervision approach is that it can deal with large amounts of unlabelled data and Mintz (2009) also states that it is relatively domain independent, since it doesn't rely on human annotation and can be trained on different types of texts. Earlier approaches often use manually labelled training data for supervised learning, for example in Zhou (2005) and Kambhatla (2004), which is not only inefficient and expensive, but also has a tendency to become quite dependent on the domain for which there is annotated data available. This makes supervised approaches unsuitable for application on a wide variety of different texts.

The improvements on the Mintz approach described in for example Hoffmann (2011) and Surdeanu (2012) make useful contributions to the problem of entity relation extraction and they all show significant improvements in the results. However, they don't address the problem of a relation being expressed in sentences where the entities don't literally occur in the same sentence. This is a complex problem, since looking over the boundaries of a single sentence would make the task much more complex. In that case all possible combinations of all entities in a document could potentially be related, and instead of learning relations from single sentences, the whole document text should be considered. Especially in relatively long documents with many entities, considering all entities in the entire document would make the problem computationally extremely complex. This thesis will explore a way to capture more expressions of relations in a text by using coreference resolution as an extension of the distant supervision method. A coreference resolution system will be used to merge entities with their coreferences. In this way the training data will not only contain the sentences where two related entities literally occur in the same sentence, but also the sentences where one or more entities are mentioned by using a coreferent. Not only does this increase the number of relations that can be found in the text, it also expands the available knowledge of the entities. Consider the following example sentences:

(3) a. The next speaker is President Obama.
    b. He is the president of The United States.

In existing distant supervision approaches neither of these sentences will be taken into account, since they both only contain one named entity. When using coreference resolution, the word 'he' will be linked to the name 'President Obama' which will turn sentence 3b into a useful instance, containing both 'President Obama' and 'The United States'. Using coreference resolution will increase the probability of finding a relation expression in the text, without needing to look over the boundaries of a single sentence, which keeps the problem of detecting related entities manageable. Using all the different types of mentions of entities in a text and treating them the same as the literal mentions, will give a lot more training data and will possibly improve the results.

This thesis is not the first to use coreference resolution in relation extraction, but it is unique in the fact that it is specifically focused on its effect. This investigation does not only give an analysis of the use of coreference resolution in relation extraction, but also gives an insight in how far this kind of complex linguistic annotation tool has developed, and whether it is ready to be used as a component of a larger system such as relation extraction.

Chan (2010) is an example of a relation extraction system that uses coreference information. The paper uses different sources of background knowledge to improve the results of relation extraction. Coreference resolution is used as a constraint, to prevent entities from participating in a relation with one of its coreferents, which would mean that the entity is related to itself. An entity that is related to itself means that there is a relation between the entity and one of its coreferents and the paper states that there are no reflexive relations in the relation extraction system. To give an example, in the sentence below it is not useful for a system to try to find a relation between 'his' and 'Bill', if it is a given that the relations the system can find does not contain relations between an entity and itself. The assumption in this sentence is that 'Bill' and 'him' refer to the same person.

(4) Bill saw that Mary was looking at him.

Even though this is a useful insight and improves the results of the relation classifier, this use of coreference resolution does not explore its full potential in the task of entity relation extraction. Rather than preventing an entity to have a relation with its coreferent, in the current thesis the entity mentions will be merged with all their coreferents. This does not only capture the problem of coreferents relating to themselves, since they will be treated as the exact same entity, but it also increases the amount of mentions for the entity the coreferent belongs to, creating more training data and increasing the chance of finding the correct relations in classification.

Shinyama (2006) describes an unsupervised approach to relation extraction, and is another example of a paper that does use automated coreference resolution as a component. However, it does not investigate the effect of the coreference resolution on the relation extraction system. It would be interesting to see the difference in results when the coreference resolution is left out, because other than creating more data, it also creates a source of errors, due to the mistakes in the resolution. Shinyama does mention this problem and states that from 10 different errors in the output of the system, 4 of them were caused by errors in the coreference resolution. Even though the state of the art in coreference resolution has improved since 2006, this of course remains a problem, and information from coreferences that are wrongly tied to a certain entity will be used as instances in the data. It remains to be seen if the gain in data weighs up against the increase in errors.

Using coreference resolution is also not uncommon in supervised approaches to relation extraction. The ACE corpus (ACE, 2000-2005) is annotated with coreferences and in Kambhatla (2004) they are treated as entity mentions, similar to the approach described in this thesis. The main difference with the coreference resolution used in Kambhatla, is that no manually annotated data will be used in this thesis, which means that automatic coreference resolution will have to be applied to the training data. As a comparison to human annotation, Kambhatla (2004) also runs the classifier on data with automatically detected entity mentions, including automated coreference resolution. The modules used in the Kambhatla (2004) paper can be found in (Florian et al., 2004; Ittycheriah et al., 2003; Luo et al., 2004). The results of this automatically annotated dataset are much worse than the manually labelled data (F-score of 25.4 compared to an F-score of 52.8), because of errors in the entity detection system and the coreference resolution. This mainly shows that automated annotation has a disadvantage over human annotation in Kambhatla's approach. However, it is not measured what the effect of coreference resolution is and what the difference in performance is when this element is added to the system, compared to using a named entity recogniser only. Automatic annotation results in more annotation errors than manually labelled data and using such annotation tools will always result in noisy data. Despite this disadvantage it has proven to be successful in unsupervised and distant supervised methods, although comparison with manually labelled data is in those cases often not possible, since a manually annotated equivalent of the training data used in those approaches is almost always unavailable. The strength of methods using automated annotation is that these systems can be trained on much larger amounts of data compared to supervised methods that use human annotation.

Even though the papers that are mentioned above do use coreference information, it remains unclear what the effect of coreference resolution on a relation extraction system is, since no comparison has been made with equivalent systems that don't use this information. This thesis will explore the use of coreference resolution in a distant supervision approach to relation extraction. The coreference resolution will be added to a reimplementation of an existing approach which is described in Mintz (2009). Additionally this thesis will provide a detailed analysis of the method described in Mintz, and will propose possible additions and changes that could be made to the system to increase its performance.

2 Background Literature

2.1 Supervised

Most early approaches to entity relation extraction were fully supervised machine learning methods, using hand-labelled data where entities and the relations between them were annotated. Kambhatla (2004) and Zhou (2005) are examples of supervised relation extraction and describe approaches that uses the The NIST Automatic Content Extraction (ACE, 2000-2005) corpus to train an SVM classifier for relation extraction. The ACE corpus is widely used in supervised relation extraction and consists of texts, annotated for entities, relations and coreferences. Both Kambhatla (2004) and Zhou (2005) explore a variety of different features, in an effort to improve the results of supervised relation extraction. Kambhatla (2004) shows that especially the use of dependency parses and other syntactic features prove to be useful. However, even though Zhou (2005) uses largely the same approach, it finds that the positive effect of syntactic features can also be achieved by more shallow methods. In addition to the features used by Kambhatla (2004), Zhou (2005) uses more shallow syntactic chunking features that do not require the data to be fully parsed. The positive effect of including syntactic features can largely be captured by these syntactic chunking features, which requires a lot less computational effort in comparison to features from full dependency and syntactic parses. The reason why syntactic chunking performs so well, is possibly because many of the relations in the ACE corpus are short-distance, meaning that the related entities are often close together, which leads to very short dependency paths between the entities. This means that the success of syntactic chunking might be related to the nature of the texts and relations in the ACE corpus, and might not be suitable for other corpora.

Even though both the investigations described above are very different from the current thesis, the findings are still very relevant. Distant supervision approaches, such as Mintz (2009), have quite a similar architecture to many supervised relation extraction systems, even though it doesn't use human annotations. Findings about relevant features for relation classifiers are important for both distant supervised and fully supervised approaches. Kambhatla (2004) and Zhou (2005) show that including dependency or other syntactic features in a relation extraction system is useful, whether they are obtained by full parses or more shallow chunking. These kind of features are now a very common component of many different relation extraction systems. The use of syntactic chunking seems promising, since it reduces the computational effort and only results in a slight drop in performance on the ACE corpus. Mintz (2009) mentions it as a possible improvement on his system. Since the coreference resolution system used in this thesis relies on full parses of all sentences, using a more shallow chunking approach would have to be done in addition to full parsing, which means that it would have no computational advantage.

2.2 Unsupervised

The unsupervised approach to relation extraction described in Shinyama (2006) tries to find entity pairs that are related in a similar way by a clustering approach. The assumption is that entities that are related in a similar way, will appear in a similar context. The results are clusters of relations, containing many entity pairs. Even though this is a quite successful approach for the extraction of related entities, it is not straightforward to compare the output with supervised and distant supervised approaches. Related entities are found, but the relation names themselves are extracted from the text, which means that there is an open set of possible relations. The advantage is that in theory any type of relation between two entities can be found, and the system is not dependent on a predefined set of relations. But on the other hand the system is not very useful if you're interested in a specific kind of relation, and it is often not possible for the system to label the relations in a way that is informative for a human. Shinyama (2006) uses the example of hurricanes and locations affected by them. The system looks at basic patterns and clusters entities where one entity (location) is hit by another entity (hurricane). However, the pattern is_hit is not informative if you don't know what kind of entities are involved. In contrast to this, the relations in Freebase are far more specific and informative. For example in the relation location/location/contains, it is immediately clear what the possible types of entities are, and how they will be related.

Banko (2007) developed an interesting unsupervised method that doesn't need parses or named entity tags to extract relations from web text. The advantage is that the system is very flexible compared to (distant)supervised approaches, because it can be applied to a very large variety of text types and doesn't need a predefined set of relations. The downside is that it is not possible to predict what relations will be found, and human judgement is needed to determine whether a found relation is informative. As mentioned, an interesting feature of this system is that it doesn't use a named entity recogniser. This means that the entities in the relations don't have to be names, but can be any noun phrase or proper noun. This can result in very general relations and relation instances. For example, an instance of the relation hired_by, could be <executive,company>.

2.3 Distant Supervised

To avoid the expenses of human annotation, Bunescu (2007) describes a system that uses minimal supervision. A small set of hand-crafted data is used to collect training data. This data consist of a small set of well-known entity pairs from which it is known that they participate in a certain relation. Sentences containing both entity pairs are extracted from the web and are used as training examples for the relation they occur in. In this approach, the set of manually collected relations and entities serve as a knowledge base. Even though in the paper the name minimal supervision is used, the approach is almost identical to distant supervision, except that in methods using distant supervision, an existing knowledge base is used. This results in a bigger and more varied dataset for training and completely removes the need for human annotation.

The relation extraction system that forms the core of this thesis, is described in Mintz (2009). Similar to Bunsescu (2007), the data is annotated by a database rather than humans, which allows a larger dataset, and avoids expenses. Instead of a hand crafted set of relations and example instances, the large database Freebase is used to annotate relations in Wikipedia texts. A logistic regression classifier is trained on this data in a similar way as supervised relation extraction systems, and different lexical and syntactic features are combined to optimize the performance. The exact architecture of this system will be described in detail throughout this thesis, since a reimplementation will be used as a baseline.

Many recent approaches to relation extraction use distant supervision as a basis, and attempt to improve the results by addressing specific problematic elements of the system described in Mintz (2009). One issue with the baseline approach is the assumption that a relation will be expressed in every sentence where the participating entities occur together. This is an overgeneralisation and results in quite a large number of false positives, since related entities can be in the same sentence while the relation is not explicitly expressed. Riedel (2010) relaxes this assumption by stating that in a set of sentences containing the same entity pair, at least one of these sentences will express the relation between the entities. In order to model this approach, the task has to be split up into two subtasks. One is finding relations between entities, the other is finding which sentences express this relation. Although this improves the accuracy of the entity relation recognition, it is a significant complication of the method described in Mintz, because a complex multi-level classifier has to be constructed in order to solve the problem.

The same holds for Surdeanu (2012) and Hoffmann (2011). They address a similar problem with the distant supervision approach and also develop a multi-level classifier instead of the straightforward multi-class classifier used by Mintz. The issue they address is the fact that the same entity pairs can occur in more than one relation, and again it is a very difficult task to determine which relation is expressed in which sentence. Entity pairs with more than one relation are quite common and finding a way to deal with this is an important development. To give an example, it is very common that a person dies in the same location as where he was born. In this case the entity pair consisting of this person and location, should be assigned both the relations people/person/place_of_birth and people/deceased_person/place_of_death. In the systems developed by Mintz (2009) and Riedel (2010), this creates a problem, because only one relation can be given to a single entity pair.

Recently, Zeng (2015) made an improvement of the distant supervision methods of Mintz (2009), Riedel (2010), Hoffmann (2011) and Surdeanu (2012), outperforming all these previous approaches. The paper follows the approach of Hoffmann (2011) in dealing with the multi instance problem, but also addresses another issue with previous relation extraction systems, which is that almost all of them rely heavily on complex linguistic annotation. And, as menioned before, the disadvantage of relying on automated annotation tools is that the resulting annotations are never perfect and are always a source of noise. Zeng (2015) creates features using neural networks rather than linguistic annotation, avoiding the use of complex NLP tools. Because of the decrease in noise caused by the avoidance of automated linguistic annotation tools, this neural networks relation extraction system outperforms the earlier described distant supervision methods.

3 Methods

3.1 Overview Mintz (2009)

Mintz (2009) describes a very well formulated relation extraction system that performs very well, especially considering its simplicity. As is described in the Background Literature, there are more recent relation extraction systems that have a better performance. The goal of this thesis is to give an analysis of the effects of coreference resolution on a relation extraction system. The relative difference in performance between the baseline and the system using coreference information is more important than achieving state of the art results. Therefore a transparent system with a reasonable performance, such as Mintz (2009), is more suitable to serve as a baseline than the more complex, high performing relation extraction system such as Riedel (2010), Hoffmann (2011), Surdeanu (2012) or Zeng (2015). This section will give an overview of the materials used in Mintz (2009), and will give a description of its architecture. The Methods and Implementation section will provide more details of the reimplementation of this system that will be used as a baseline.

3.1.1 Materials Mintz (2009)

The core idea of distant supervision is that training data can be supervised by a database containing knowledge, rather than by human annotations. The Mintz (2009) paper uses the Freebase data from July 2008 and this exact same data will be used for the current thesis. At that time Freebase contained about 9 million different entities and 7300 different relations, but they are filtered in the following way:

> We next filter out nameless and uninteresting entities such as user profiles and music tracks. Freebase also contains the reverses of many of its relations (bookauthor v. author-book), and these are merged. Filtering and removing all but the largest relations leaves us with 1.8 million instances of 102 relations connecting 940,000 entities.

(Mintz et al., 2009)

In distant supervision, it is important that there is a close relation between the text and the database. The texts will be annotated using the knowledge in the database, so it is essential that the entities that are present in the database, can be found in the text. The quality and the amount of annotated data is essential for training a relation extraction system and if the difference between the database and the text is too big, this is not possible to achieve.

This is why Wikipedia texts are very suitable when Freebase is used as a database, because Freebase is for a big part constructed from Wikipedia infoboxes. 800000 Wikipedia articles are used for training, and 400000 for testing. The Wikipedia data is tokenized by Metaweb (2008) and annotated with Stanford coreNLP (Manning et al., 2014) for Penn Treebank POS-tags (Marcus et al., 1993), NE-tags (person, location, organisation and miscellaneous) and MINIPAR (Lin, 2003) dependency trees.

Mintz claims that the distant supervision approach is more domain independent than supervised approaches. The reason for this is that distant supervision is not dependent on human annotations, and is not restricted to the texts for which such information is available. But even though distant supervision is not dependent on human annotated texts, this does not mean that it can be easily trained on any type of text. As mentioned earlier, it is essential that there is a close similarity between the database and the text. Therefore it will only be possible to achieve good results when a system is trained on texts for which there is such a database available.

3.1.2 Architecture Mintz (2009)

The Mintz (2009) relation extraction system is a logistic regression classifier, trained on 800000 Wikipedia documents, annotated with the filtered Freebase relations. The distant supervision assumption plays an important role in the baseline implementation. According to this assumption, the relation between two related entities will somehow be expressed, if those entities occur together in the same sentence. Every occurence of two related entities appearing in the same sentence will be considered a training instance for the relation they occur in. In this way the texts will be annotated for entities and relations, using the knowledge of Freebase. The result is a dataset that is automatically annotated. Even though the quality of the annotations will not be as good as human annotations, a supervised learning algorithm will be used to train the classifier. This makes the process of distant supervision relatively similar to full supervision.

Once the text is annotated, the next step is to extract features from the sentences containing related entities. Mintz (2009) uses both lexical features and dependency features. The lexical features consist of the named entity tags of the entities, and the words and part-of-speech tags of the tokens between the two entities and surrounding words. For the dependency features, every sentence is parsed by the dependency parser MINIPAR. These features contain the dependency path between the two entities and a window on the left and right of the entity pair. In this way, the system will learn to recognise related entities, by learning the lexical and dependency context of related entities. The features are extracted for every sentence containing an entity pair that is an instance in Freebase. Whenever a sentence contains two or more entities, features will be extracted for every combination of two entities that form an instance in Freebase.

This is a very broad overview of the architecture of the system described in Mintz (2009). For the purpose of this thesis, the system is reimplemented to serve as a baseline. The exact details of the system will be discussed throughout the rest of the thesis.

3.2 Coreference Overview

In order to be able to measure the influence of coreference information on a relation extraction system, a system will be implemented where this information is used as an addition to the baseline implementation. The coreference information consists of clusters of entity mentions, that all refer to the same entity. Where in the baseline ’Obama’ and ’Barack Obama’ would be considered two different entities, using coreference cluster information will make sure these will be considered the same person. Beside different spellings for the same name, these coreference clusters also contain other elements, such pronouns and other noun phrases that refer to a certain entity. By replacing the coreferents with the name they refer to, more mentions of the entities can be extracted, which will lead to more training data.

3.3 Materials

3.3.1 Preprocessing

The original data, recieved from Mintz, was already annotated as described in Mintz (2009), so no further preprocessing was necessary. For the baseline implementation, the original annotated data from Mintz (2009) will be used. However, in order to achieve optimal results from the coreference resolution, it is necessary to have up to date annotations. Therefore the entire dataset has been reparsed using the Stanford CoreNLP package (Manning et al., 2014). This section will give a description of the part-of-speech tagging, named entity recognition, dependency parsing and coreference resolution. Lemmatization is a necessary component for the coreference resolution, but will have no other use in the implementation of the system.

It is unfortunately not possible to measure the performance of the automated NLP tools on the Wikipedia data used in this thesis, because there is no gold standard available. The results reported in the respective papers of the different annotators will probably not be achieved on the dataset used in this thesis. However, Wikipedia texts largely consists of well-formed sentences, as opposed to, for example, online chat data. The assumption is that the performance of CoreNLP annotation tools on Wikipedia data will not be problematic.

POS Tagging

The part of speech tagging system that is used in the CoreNLP package is described in Manning et al. (2003), and annotates all tokens in a document with Penn Treebank part-of-speech tags. The accuracy reported for this pos tagging system is 97.24%, outperforming the state of the art at the time the paper was written. The essential component of the tagger is the so-called bidirectional model, that uses both information from the previous tag and the following tag to determine the most probable tag for every word in the sentence. It also makes use of lexical features, that make sure the surrounding words are properly taken into account and it has a well-developed system for dealing with unknown words. This is particularly useful for texts from a different domain than the tagger is trained on, since these texts will have relatively many unknown words for the system. This is a very relevant property for the application of the tagger on the Wikipedia data used in this thesis.

Named Entity Recognition

CoreNLP provides a linear chain CRF named entity tagger for which the precise method, according to the CoreNLP website[1], is not described in any paper, although the paper for citation is Finkel et al. (2005). The named entity tags that will be used in this thesis are Person, Location, Organization and Miscellaneous.

This section will briefly explain the basics of a linear chain CRF tagger (Lafferty et al., 2001) and will provide an overview of the additions that Finkel et al. (2005) makes to the existing CRF named entity recognisers at the time. The assumption is that this forms the basis for the implementation in coreNLP.

A conditional random field model (CRF) is quite similar to a Hidden Markov Model (HMM) in the sense that it computes the probability of a hidden tag sequence based on some observations. But where an HMM

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

tagger is limited to using so-called emission and transition probabilities, a CRF could in principle model any kind of information to calculate the probabilities. A linear chain CRF uses information only about the current and previous word in the sentence, but in principle a CRF can use any kind of information. A linear chain CRF named entity recogniser uses information about the previous, current and next word in order to determine the most likely NE tag of every word in the sentence. It doesn't only look at the tokens themselves, but can use many different features, such as POS tags and ngrams.

The most important addition of Finkel et al. (2005) is that it models certain non-local information about entities in a text. Earlier approaches only look at the information present in the current sentence when choosing the right named entity tag for a certain word, while there may be clues in other places in the text that indicate which tag could be correct. For example, when more than one tag is competing for a certain word, it could be useful to look at how this word has been tagged in the past, because in many cases the correct tag of the word will be consistent with this. This information can't be found in the local (sentential) information considered by the linear chain CRF model only. By modelling this kind of non-local information and by using an alternative optimization algorithm than Viterbi (Gibbs sampling), the resulting NE tagger outperformed the state of the art at the time with an f1-score of 86.86.

Dependency Parsing

The basis of the CoreNLP dependency parser is the transition-based parser described in Nivre (2004). Nivre describes an incremental parsing algorithm that uses a set of simple transition rules to construct dependency parses[2]. The parser annotates the data with Universal Dependencies (Nivre et al., 2015).

The CoreNLP dependency parser (Chen & Manning, 2014) differs from traditional transition based dependency parsers in that it uses a neural networks classifier. In earlier approaches, the features are often not only less than optimal, but are also computationally expensive. By using neural networks, this parser does not only outperform the state of the art transition based parses with an accuracy of 90.7 %, but also speeds up the process significantly. It achieves a speed of 1013 sents/s as opposed to 560 for the Malt parser (Nivre et al., 2006).

[2] A tutorial on transition based dependency parsing can be found at http://stp.lingfil.uu.se/ nivre/docs/eacl3.pdf

The dependency parses are both an essential component for the coreference resolution, as well as the feature extraction of the relation extraction classifier.

Coreference Resolution

The papers that describe the CoreNLP Coreference system are Raghunathan et al. (2010), Lee et al. (2011), Lee et al. (2013), Recasens et al. (2013) and Clark & Manning (2015). The first four describe the architecture and performance of the deterterministic coreference system and the most recent one describes the statistical coreference resolution. This section will describe the essential parts of these papers, starting with the deterministic approaches. The model used in this thesis is the statistical coreference resolution and will be discussed later in the section. Even though the deterministic coreference resolution is not used in the current thesis, it forms an important background for the statistical coreference resolution.

Raghunathan et al. (2010) describes an at the time novel system for coreference resolution, using a relatively simple deterministic, unsupervised algorithm. It uses a sieve-based approach, adding tiers of coreference rules one at a time, as opposed to models applying all the models in a single function. The advantage of this system is that for every tier, the information gained by the previous tier is available, which makes it possible for the model to make more informed decisions at every step. The tiers are applied in order from high precision to low precision, so that strong features are given precedence and weaker features will be able to use the information acquired by high precision models. The system has a total of seven passes, that will be explained shortly here below.

Pass one clusters mentions that consist of exactly the same text, which has of course a very high precision, because it will only cluster identical names. The second pass consists of a set of different conditions. In order for two mentions to be considered coreferents, one of these conditions has to be satisfied. An example of such a condition is if one of the mentions is an acronym of the other. The third pass is less precise and matches two mentions that have the same head word. Since it is not uncommon for two different entities to have the same head word ('Gothenburg university' and 'Chalmers university'), a set of conditions is defined to prevent these kind of mentions to be linked. Two mentions will only be linked if they match all of the conditions. The passes 4 and 5 relax some of the constraints from pass 3, to include more mentions with the same head. In pass 6 two entities will be linked if the head matches any other word in the candidate mention and the last and 7th pass deals with pronoun resolution. Applying the pronoun resolution as the last step is important, because it has to be able to use all the agreement information available in the earlier constructed clusters.

An advantage of this sieve based approach is that it is relatively easy to extend it by adding tiers, which is what Lee et al. (2011) does by adding five additional tiers. The first addition is a mention detection sieve, that first extracts all candidate mentions from the text, before applying the actual coreference system. This sieve is a separate step before the coreference tiers and is designed to have high recall, to make sure as many mentions as possible are captured. Another tier that is added is the 'discourse processing sieve'. This sieve detects speakers by, for example, taking the subject of verbs similar to 'say'. The sieve then uses a set of simple rules, to identify coreferents. An example of such a rule is that all the mentions of 'I' by a certain speaker are coreferent with both each other, and the speaker itself. After the string match sieve described in Raghunathan et al. (2010), Lee et al. (2011) also introduces a relaxed string match sieve that matches strings that are identical after removal of all the text after the head word. This is necessary to cluster entity mentions that have a relative clause attached to it. Another addition is proper head word match that matches two entities with the same head word, if this head word is a proper noun and satisfies a set of extra conditions. The alias sieve uses information from different databases (Wikipedia infoboxes, Freebase and Wordnet synsets (Miller, 1995)) to find mentions that are marked as aliases in the database information. The lexical chain sieve links mentions together that are linked through hypernymy and synonymy relations in WordNet.

Lee et al. (2013) discusses the architecture and performance of Raghunathan et al. and (2010), Lee et al. (2011) in detail, but doesn't make any big modifications or additions to the system. Recasens et al. (2013) describes a system for singleton detection, which output improves the coreference system. A singleton means an entity mention that doesn't have any coreferences. Identifying this distinctions between the mentions in a text before applying a coreference system, will prevent mentions that are most likely to be a singleton to end up in some coreference cluster and prevents coreferent mentions to end up as a singleton.

The systems described in this section so far are the ones used in the deterministic coreference resolution of the corenlp package. The system used for annotating the Wikipedia data of this thesis is the statistical coreference system described in Clark & Manning (2015). Even though the architecture of statistical coreference is quite different to what is described above, it is still useful to know how they work, because some of the core ideas behind the sieve based approach prove to be useful in statistical coreference resolution as well, as is described in Clark & Manning (2015). The first similarity is the idea of building up the coreference clusters incrementally, so that later stages can profit from the previously acquired knowledge. The second similarity is to use features from the entire entity cluster instead of just a mention pair, which is specifically useful for pronoun resolution. Clark & Manning (2015) gives a good example of why this last property is important. If there is a cluster containing the entity mentions 'Clinton' and 'she', it is very useful if the gender information of 'she' is included in the decision whether the mentions 'Clinton' and 'Hillary Clinton' are coreferent, or 'Clinton' and 'Bill Clinton'. If only the features of the pair under consideration would be taken into account, Clinton wouldn't have any gender information and could be linked to either of those entities.

Clark & Manning (2015) uses a statistical approach rather than the earlier described deterministic sieve approach. The first stage of the program is to predict whether pairs of mentions are coreferents. In a later stage, coreference is predicted between mention clusters, rather than mention pairs. In this way different clusters are merged until all clusters represent a different coreferent set. The input for the cluster merging algorithm is the output of the mention pair merging algorithm. To reduce the search effort, the mention pairs are ranked from very likely to very unlikely and pairs with a probability below a certain threshold are not considered for merging. The reported f1-score for the statistical coreference system used in this thesis is 56.05.

According to the Stanford CoreNLP website[3], all Stanford CoreNLP coreference systems are tools for nominal and pronominal coreference resolution. The systems are limited to making coreference links between named entities (Barack Obama), noun phrases (the president) and pronouns (he).

3.3.2 Freebase Data

As mentioned in the Overview Mintz (2009) section, the filtered Freebase data from July 2008 will be used for both the baseline and the system that includes coreference information. Half of the instances will be used for annotating the training data, and the other half for the testing data.

In addition to the manual filter that has been performed by Mintz (2009), the Freebase data is filtered again for the purpose of this thesis. Even though Mintz claims to have used the 102 largest Freebase relations, surprisingly there are still some relations included that have only one or two

[3] http://stanfordnlp.github.io/CoreNLP/coref.html

instances. Since a machine learning system will never be able to make a good judgement based on a single instance, these relations are removed from the set.

Apart from this, the entities in the Freebase relation instances can be a great variety of different types of entities. There are persons, book titles, locations, but also strategic position of sports players on the field and many other things. Since the named entity recogniser used in this thesis is restricted to find persons, locations, organisations and a miscellaneous class, the classifier will never be able to properly learn relations that contain different entities than those four. Including relations that the named entity recogniser can't identify, means that these relations will only be part of the training data when the named entity recogniser makes a mistake. It can for example happen that by some mistake, a disease is tagged as being an organization, and the sentence in which this occurs also contains the name of a person. It is possible that this sentence will be considered by the system as an expression of the /people/deceased_person/place_of_death relation. The classifier will then learn that this relation is one that typically occurs between a person and an organization, which is of course not true. Including relations that the named entity recognizer can never find only results in classes that the classifier will never be able to reliably identify.

All relations that contain entities that can not be found by the named entity recogniser, and relations that only have one or two members in Freebase, are removed from the set of relations. This results in a re-filtered set of 25 relations. All relations, both the original and the re-filtered set, can be found in appendix A. Every system described in this thesis, will be trained and tested on both sets of relations, and the results will be compared.

3.3.3 Wikipedia Data

The Wikipedia texts used in all systems described in this thesis is identical to the texts used in Mintz (2009). It consists of 800000 documents for training, and 400000 documents for testing. The documents are stripped in such a way that only the text remains. The structured data that can be found in Wikipedia, such as links and infoboxes are all removed from the documents. The relation extraction system will be entirely based on the text of the documents, and the knowledge in Freebase.

3.4 Evaluation Metrics

To test the effect of coreference resolution on a relation extraction system, and to analyse the performance of the system in general, two different evaluations will be presented in this thesis.

The first evaluation will be an automated evaluation. From both the Freebase data and the Wikipedia text, a test set is held apart. The Wikipedia data will be annotated with the Freebase data to create a set of test instances. The Freebase relations will serve as a gold label. All systems will classify the testing instances and from this a precision and recall value can be calculated. None of the testing instances will have occurred in the training set, whether they are related or a negative instance. This automated evaluation will make it possible to see the relative difference in performance between the different relation extraction system.

In order to give a concrete insight in the behaviour of the relation extraction systems, a small manual evaluation will be performed. Entity pairs will be extracted from two Wikipedia documents that are not in the training set. The relation extraction systems will then classify these entity pairs. This evaluation is too small to properly estimate the performance of the systems, but in combination with the automated evaluation, it will give a detailed view of the behaviour of the relation extraction systems that are implemented in this thesis.

4 Implementation

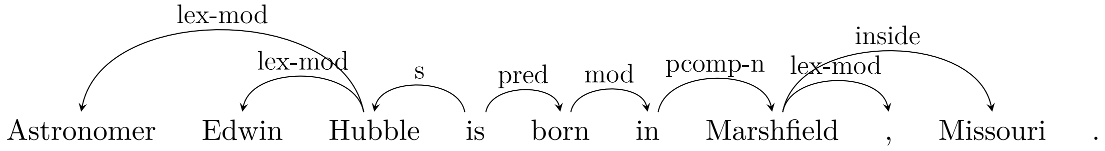
This section will give a detailed description of the design and implementation of the relation extraction system. It will go through the components of the system one by one and will explain the structure of each part. Where present, it will also discuss possible difficulties and problems. The aim is to follow the baseline implementation as described in Mintz (2009) as close as possible. The original code of the baseline is not available, and therefore the details of implementation might differ.

4.1 Entity Chunking

Entity chunking is a necessary step before the data can be annotated, which means that names that consist of multiple words need to be considered as one word. Since the boundaries of the names are not annotated in the data, with for example B-O-I tags, the maximum sequences of words with the same entity tag are chunked together. After this chunking process, all names will be treated as a single word.

One problematic aspect of this chunking, which is not thoroughly explained in Mintz (2009), is that this collapsing of entities affects the dependency tree of the sentence. In the dependency trees, last names are often the head of first names. Simply collapsing the dependency tree would in this case result in entities being their own heads. To illustrate this, let's consider the example sentence that is given in Mintz (2009):

(5)



In this sentence, there is a dependency arc between Edwin and Hubble. Merging the two entities together means that this will turn into an arc between Edwin Hubble and Edwin Hubble. It is not clear whether Mintz keeps these reflective dependency arcs in his implementation. However, for reasons explained in the feature extraction, these arcs are removed in the current thesis, to prevent the name itself turning up as part of a feature.

In the implementation of the entity chunking, the indexes of the dependency arcs need to be adjusted. For example, the word 'born' previously was the 5th word in the sentence, but since the name Edwin Hubble is now considered one name, it suddenly becomes the 4th word in the sentence. This can be easily dealt with by extracting the difference in length between the old and the new sentence from every index after the merged name.

Every entity chunk inherits all the dependencies and the POS tag from the head of the name.

4.2 Data Annotation

All Wikipedia texts in training and testing are annotated with the relations in the Freebase data. One half of the Freebase data is used to annotate the training set, the other half is used for the testing set. In every sentence of the training data, the entities and the relations between them are annotated. The same holds for the testing data, except that for every entity pair it is also checked whether the same pair is also part of the training data, to prevent the same entities occurring in both the training and testing data. From all entities that don't occur in Freebase, 1% is taken as negative data and will be considered as examples of entities that express no relation. Even though this results in false negatives, the effects of this are assumed to be minimal because of the close relation between Wikipedia and Freebase. Freebase is a very large database that is partly constructed of Wikipedia data, so many of the related entities present in Wikipedia can be found in Freebase.

In Mintz (2009) it is not explained how to deal with entity pairs that occur in more than one relation. The decision made in this thesis, is that the features for each entity pair will be considered training data for each relation it occurs in. This is not an ideal solution, but without changing the distant supervision assumption (every sentence with a related entity pair will express the relation), it is the best alternative. Considering the design of the baseline it is not possible to tell which mention of such an entity pair belongs to which relation, so therefore the generalization is made that every mention belongs to every relation. The disadvantage of this solution is that it creates false positives, which means that because every mention of such an entity is considered training data for every relation, there will be quite some mentions that will be used as training data for a relation that is not expressed in that specific sentence. The assumption is that this effect will be filtered out, because of the large amount of training data. In testing, an entity pair that has more than one relation can never be classified 100% correctly, because the system doesn't allow multiple labels per instance. Each entity pair will be tested for each relation it has in Freebase. For example an entity pair that has two different relations will never be able to achieve a higher accuracy than 50%.

The fact that the data will contain false positives is a problematic aspect of the Mintz approach in general. Even if every entity pair would only have one relation, this does not necessarily mean that this relation is expressed in every sentence both entities occur in. The systems of Surdeanu (2012) and Hoffman (2011) are able to deal with multiple labels for an entity pair, but their systems have quite a different architecture than the system described in this thesis.

4.3 Feature Extraction

This section will describe and discuss the different lexical and dependency features used in both the baseline and the coreference implementation of the relation extraction system. Important to note is that the relation extraction system will classify entity pairs, and not the individual mentions of the entity pairs. This means that the features of all entity pair mentions will be combined into one feature vector for every entity pair.

All features are a conjuction of a number of different elements, resulting in a very large vocabulary of different features. The reason for this is that this results in a performance with high precision and low recall, which is preferable for a relation extraction system. To illustrate this, you rather want that the relations found by the system are correct (high precision), than that as many relations as possible are covered, but with a higher error rate (high recall).

4.3.1 Lexical Features

The lexical features contain information about the words, the NE and POS tags of the related entities, and the sentences they occur in. Each lexical feature is a conjunction of four different elements: the named entity tags, a middle part, a window feature and a boolean value.

The first element of the lexical features are the named entity tags of the names in the entity pair. In the original annotations of the Wikipedia data, the named entities that can be recognised are: location, person, organization and miscellaneous. Even though in the reparsed data there is a bigger variety of named entity tags, only those four are considered.

It's important to note that the names themselves are not part of any feature, preventing the program to have a bias towards the names it is trained on. This ensures that the system can deal with new names relatively easily. Although this doesn't mean that there is no bias at all for common names or famous people that occurred in the training. Some names will often occur together with specific words, that will become a part of the features for these names. These words don't necessarily have anything to do with the relation between the entity pairs, but rather with one of the names participating. For example the name Obama and his place of birth will probably occur a few times throughout the corpus. It is very likely that the word 'president' ends up in the features of those sentences, which creates a situation where the system is better at finding the place of birth of people who are president. Considering the large amount of training data, these effects will hopefully filter out.

The second feature that the combined lexical features contain is called the lexical middle. These are all the words between the two entities and their simplified part of speech tags. The simplified part of speech tags only contain 7 tags: Noun, Verb, Adverb, Adjective, Number, Foreign and a class named Closed for every remaining class. This is done to prevent the vocabulary of different features from expanding too much.

There is also a lexical window feature containing k words to the left of the first name and k words to the right of the second name in the sentence. There will be one lexical feature for every k from 0 to 2, resulting in three lexical features for one sentence.

All instances in Freebase have an order in which the entities appear. For example, in the relation location/location/contains, the first entity always contains the second. All lexical features have a boolean value indicating whether the entities appear in reversed order in the sentence. To illustrate why this is important, we look at the example sentence again. The entities Missouri and Marshfield occur in the relation location/location/contains. It is important that the construction 'Marshfield, Missouri' is an indication that the second entity contains the first and not the other way around. Some relations, like /people/marriage/spouse are symmetric, so if <John,Mary> belongs to this relation, it means that <Mary,John> will be a member too and there is no difference in meaning between the two. But more often, the relations are directional, and the feature that indicates the order is necessary.

The example below shows the lexical feature of the example sentence for k=1. Note that all these elements form a single lexical feature. This example is based on the example features that can be found in table 3 in Mintz (2009).

inverse	left	NE1	middle	NE2	right
False	Astronomer	P	was/VERB born/VERB in/CLOSED	L	,

Table 1: example lexical feature

4.3.2 Dependency Features

The dependency features are a conjunction in the same way as the lexical features, containing the named entity tags of the relevant names, a middle part and a window to the left and the right. Even though the named entity tags are not a dependency related feature, they are still included in the dependency features.

The most complex part of the dependency features is the middle. This is the dependency path between two entities, constructed in a similar way as in Snow (2005). The first step in getting this dependency path is to treat the dependency tree as an undirected graph. The direction of the graph's edges are still stored, but don't constrain the direction in which the path can be followed and can be seen as a part of the label of the graph. The dependency path is the shortest path in this graph, with a maximum of 4 edges between the entities. The implementation used in this paper uses a simple backtracking algorithm to find this shortest path[4] ,

[4] The code for finding the shortest path is based on the code from https://www.python.org/doc/essays/graphs/

terminating when the shortest path is found, or when all possibilities exceed the maximum of four edges, or in the rare case where the dependency tree is ill-formed and there is no path between the entities whatsoever. The resulting dependency path contains the direction of the edges, the relation labels of the edges and the words they connect, excluding the names of the entity pairs themselves. To go back to example 4, the resulting dependency path between Edwin Hubble and Marshfield is the following:

(6) ← s was → pred born → mod in → pcomp-n .

The dependency path between Marshfield and Missouri connects the two entities directly, which means that it only contains the arch itself: → inside. It is not clear what decision Mintz made on this aspect, but in the thesis implementation, there are no dependency features if there is no dependency path found, rather than an empty dependency feature. This means that if no dependency path is found, only lexical features are considered for that sentence.

The dependency window features are the k-th word to the left of the first entity, connected by a direct dependency edge and the k-th word on the right of the second entity, connected by a direct dependency edge. Instead of having one window length k for both the left and right window, the left and right windows have their own length k_left and k_right from 0 to 2. this means that there are at maximum 9 dependency features for every sentence, for all combinations of k_left and k_right. If there is no edge found for a value of k_left or k_right (except of course for 0, which is always empty), no feature is added.

The example below shows an example of a dependency feature, again the example is based on table 3 in Mintz (2009). The feature is the dependency feature for k_left=1 and k_right=1.

left	NE1	middle	NE2	right
Astronomer → lex-mod	P	← s was → pred born → mod in → pcomp-n	L	→ lex-mod ,

Table 2: example dependency feature

4.4 Classifier

Both training and testing is implemented using the scikit-learn (Pedregosa et al., 2011) python library. The baseline as described in Mintz (2009) uses a logistic regression classifier lbfgs with gaussian regularisation. The reimplementation in this thesis uses the LogisticRegression[5] classifier from scikit learn with equivalent settings, in order to keep the reimplementation as close as possible to Mintz (2009). Additionally, the parameter class_weight is set to 'balanced', which is recommended for classifiers where the frequency of the different classes differs a lot. Since the number of mentions of an entity pair can differ a lot (sometimes there will be only one mention, but well know entity pairs can have many), the number of features that are extracted for that instance will differ also. It is therefore necessary to add the scikit-learn Normalizer as part of the pipeline.

Following Mintz, this classifier is a multiclass classifier, which means that for each instance, only one class can be assigned. This is a problematic aspect of the baseline, since it results in an inability of dealing with instances with multiple relations. Hoffman (2011) and Surdeanu (2012) describe possible solutions for this problem.

4.4.1 Training

For training, all the instances from the training set are collected, an instance being all the mentions of an entity pair combined. From every sentence, the features are extracted and vectorised using a CountVectorizer from scikit learn. The features of every mention of the entity pair will be appended to the feature vector of that instance.

For entity pairs that have multiple relations, the extracted features will be considered training instances for all the relations they occur in.

4.4.2 Testing

For the automated evaluation there is a very similar approach. Features are extracted from all sentences of all instances in the test set, and combined into one feature vector per entity pair. The Freebase relation of the instances will be treated as gold labels. Instances with multiple labels are particularly problematic in the automated evaluation. These instances are duplicated for as many relations as it has, and will appear in the testing data once with every relation it has as a gold label. Since the feature vector for every of those instances will be identical for every relation it has, the classifier will be unable to classify all those instances correctly. And instances with two relations will be classified with a 50% accuracy at most.

[5] for comprehensible information about LogisticRegression, as well as the Normalizer and CountVectorizer, see http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html, http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html and http://scikit-learn.org/stable/modules/generated/sklearn.feature$_e$xtraction.text.CountVectorizer.html

4.5 Coreferences

The coreference output of Stanford CoreNLP consists of clusters of coreferent entity mentions for every document. Each cluster has a representative mention, which is typically the longest member of the cluster. The coreference clusters are not limited to named entities only, but can also be noun phrases and pronouns. In this thesis, coreferents are replaced by their representative mention only if the representative mention is a named entity.

Two versions of the relation extraction system with coreference information will be tested and compared with both the baseline and each other. In the first version, all the coreferences are identified and replaced immediately, resulting in a sentence with only the representative names. One of the properties of this system is that even coreferents that are not part of the selected entity pair are replaced. The expectation is that this has the effect that the vocabulary of different features will be even larger than it already is by the baseline design, because many pronouns will be replaced by the names they refer to. This results in more variety in the features. Consider for example the following sentences:

(7) a. John knows she lives in London.
b. Bill knows she lives in Paris

Now let's assume the entity pairs under consideration are respectively John and London, and Bill and London. In sentence 7a the word 'she' refers to Mary, but in sentence 7b, 'she' refers to Kate. Where in the baseline system the features for the first and second sentence are identical, in this system using coreference the features differ, because they will contain the names instead of the pronoun. Since the vocabulary of different features is already very large in the original system, increasing it in this way does not seem preferable.

This increase in vocabulary is the reason for creating a second coreferent replacement system, where coreferents are only replaced if they are part of the entity pair under consideration. In this case, 'she' will be unchanged in both sentences, unless its representative is part of the entity pair under consideration.

All replaced coreferents inherit the POS tag from the representative mention and the dependencies of the head of the coreferent. In the case that multiple words are replaced with one single word, the dependency tree needs to be adjusted in the same way as with entity chunking. The difference in length between the replaced and the original sentence will be extracted from every index after the target word to make sure all dependency arcs point to the right element.

Note that even in the case of replacing nouns and pronouns, the resulting sentence can sound strange and even ungrammatical. This for example happens in the case of possessive pronouns, where 'his book' can be turned into 'Obama book'. But since the actual text of the named entities are not part of the features, this will not have any effect on the features for these sentences. Because the context of replaced entity pairs can look different from normal entity pairs, the expectation is that adding coreference information to the relation extraction system will increase the feature vocabulary, regardless of which replacement algorithm is used.

5 Results and Evaluation

This section will present the results of the automated evaluation and will discuss the performance of all relation extraction systems that are implemented for this thesis. It will also describe a detailed analysis from the manual evaluation of two documents.

5.1 Automatic Evaluation

The test part of the Freebase data is used to annotate the test documents from Wikipedia. The Freebase relation for each instance will serve as a gold label in this automatic evaluation. Just as in the training data 1% of the entity pairs are taken as negative samples. This means that entity pairs that are not in Freebase will be considered unrelated, which is of course not necessarily true. All related entity pairs that the systems find in the test documents and the negative sample will form the test instances. The precision and recall values will be calculated according to the number of test instances for every system. The design of the automatic evaluation in Mintz (2009) is not clearly described. Therefore the evaluation in this section is an approximation of the evaluation in Mintz (2009).

The design of this evaluation gives a rather negative impression of the relation extraction system, because there will be cases where an entity pair is correctly classified for a certain relation, but if the pair is not present in the testing half of Freebase, this will not be recognised. Even though this is a downside of this evaluation method it is not problematic for the current investigation, because the focus of this thesis is the relative difference between the different relation extraction systems.

This section will compare the performance on all different relation extraction systems. For every system, the test set is processed according to the respective method. To be more specific, the baseline is trained on the original annotations used in Mintz, and therefore the test set has to be annotated in the same way in order to be able to test the system. The same counts for the reparsed baseline implementation. This means that there is a difference in the number of testing instances between the baseline and the reparsed baseline. Similarly, for the two coreference systems, coreference information is used in the same way in the training and testing set. Not only to be fully able to show the advantages and disadvantages of those systems, but also to provide an optimal test set for each relation extraction system. If the systems using coreference would be tested on a set where this information is not available, the system would not have an optimal performance.

The negative class (the 'no relation' class) is removed from the classification reports, and is not considered in the overall precision and recall values. The reason for this is that this class forms the majority and including it in the values gives a wrong impression of the performance of the system. To illustrate this, a system that never finds any relation will result in a quite high score for the negative class, and since this class forms the majority, the overall performance will seem reasonably good, even though a system like that fails completely as a relation extraction system.

5.1.1 Full Set of Relations

system	precision	recall	f1-score
baseline	0.61	0.36	0.45
reparsed	0.56	0.37	0.44
coref1	0.57	0.37	0.44
coref2	0.57	0.37	0.45

Table 3: Precision and Recall for all systems, full set of relations

This section will discuss the performance of the different systems trained and tested on the full set of relations, identical to the one used in Mintz, and will highlight some of the things that stand out.

The first thing that shows in this evaluation, is that even though the system theoretically only has a proper chance at finding instances of the 25 relations that contain persons, locations and organizations, the test set contains instances of 66 different relations for the baseline. For 30 of those

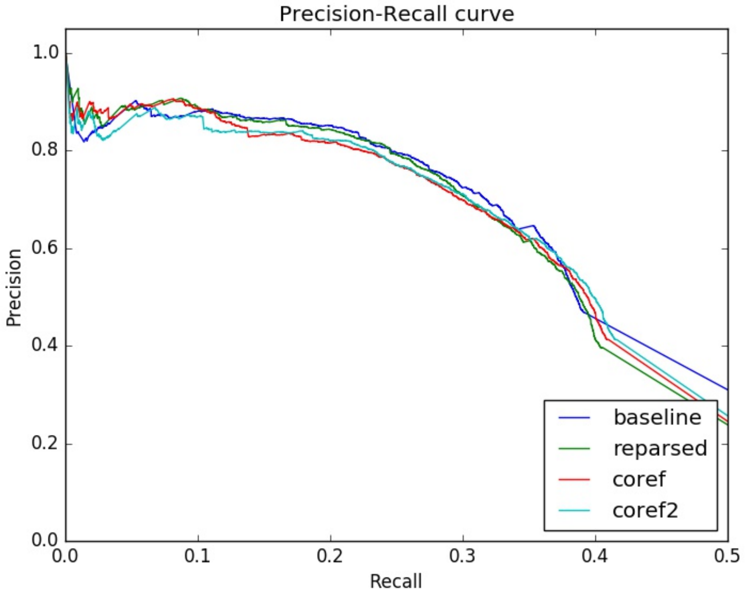


Figure 1: Precision-Recall Curve for the full set of relations

relations, no instances are correctly classified, which suggests that the baseline is indeed not capable of recognising certain relations. 25 of those 30 relations are indeed relations that are removed in the refiltered set of relations. From the remaining relations, there are 12 relations that have an f-score lower than 0.10, of which 7 again belong to the group of relations that are considered irrelevant. These numbers confirm the motivation for the decision to train and test the system on the filtered set of relations as well, to see whether the overall performance changes.

A common way to present the performance of different relation extraction systems is a precision recall curve, as shown in figure 1. Rather than presenting a single number for precision and recall, this plot shows how the system perform over different recall levels. This makes it easier to compare the performance of the different systems in a visual way. Both figure 1 and table 3 show that there is barely any difference between the different systems at all, which means that reparsing the data and adding coreference resolution doesn't seem to have an effect on the precision and recall of this relation extraction system. The curve only gives an overall impression of the performance of the entire system and doesn't show possible differences between the classes, or other variations that might influence the behaviour of the systems. Therefore it is important to also look at the performance of each system individually.

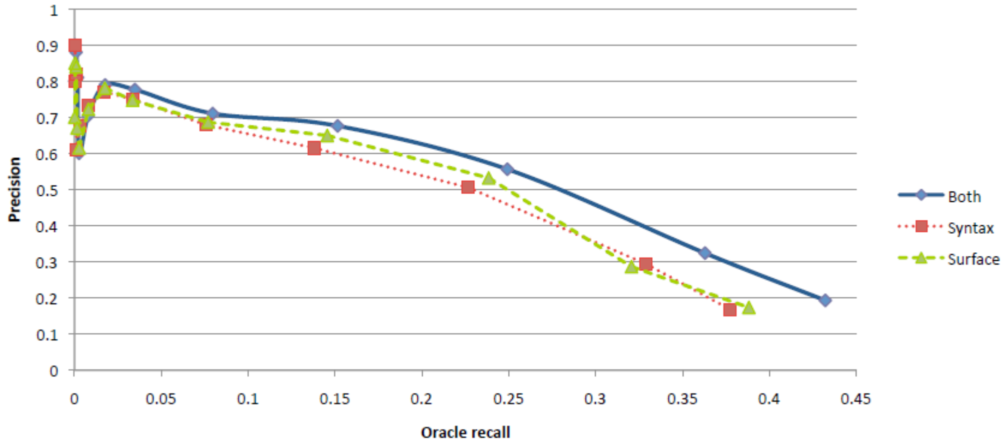


Figure 2: Precision-Recall Curve as presented in Mintz (2009)

In the results of the full set of relations, it stands out that in each system there is a small number of relations where the system performs quite well, and quite a large number where the numbers are much lower. Even in the top 5 performing relations for each system (table 4-7), the differences between the values are quite big. The following subsections will present the results of each system separately, and will conclude with a discussion about the confusion matrix for the filtered set. A confusion matrix for the full set is not provided, because the number of instances would make such a table unreadable.

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.83	0.80	0.81	148
/people/person/place_of_birth	0.68	0.54	0.60	2299
/location/location/contains	0.79	0.47	0.59	11744
/geography/river/basin_countries	0.73	0.33	0.45	364
/tv/tv_program/genre	1.00	0.25	0.40	4

Table 4: Top 5 performing relations of the baseline, full set

Baseline Table 4 shows the top performing relations of the baseline implementation. The first thing that stands out is the wide range in the number of instances for these relations, varying from 11744 to only 4. It is not simply the case that frequent relations perform best. For example the relation /people/person/nationality has a total of 3346 instances, but only reaches an f-score of 0.19, while the best performing relation has 148 instances and an f-score of 0.81. The member of the top 5 for the baseline that stands out most is /tv/tv_program/genre with only 4 instances and a precision of 1. Apparently the classifier only predicted this class once out of the 4 cases where this was the correct prediction. This means that there must be some very strong feature for this class, that is not associated with any other class. Even though the precision is high, it can't be said that the relation extraction program is good at identifying this class, because in 400000 documents, it has only been able to identify 4 instances of a television program and its genre. It is very likely that Wikipedia contains many more examples of this relation, but the named entity recognizer is not able to consequently recognise tv programs and genres.

In total, the baseline finds 21641 relation instances, with a precision of 0.61 and a recall of 0.36. This supports the statement of Mintz that the design of feature extraction creates a high precision, low recall performance. Figure 2 is the precision recall curve presented in Mintz (2009). The blue line shows the curve of the system that is used as a baseline in this thesis. This curve shows a slightly worse performance than the baseline, which could be due to differences in the evaluation design. Despite the differences, the curves of figure 2 and 1 are not very far apart and the baseline is assumed to be an accurate reimplementation of Mintz (2009).

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.76	0.76	0.76	149
/location/location/contains	0.74	0.54	0.62	9987
/geography/river/basin_countries	0.67	0.57	0.62	285
/people/person/place_of_birth	0.60	0.31	0.41	2546
/aviation/airport/serves	0.24	0.36	0.29	143

Table 5: Top 5 performing relations of the reparsed baseline, full set

Reparsed The reparsed dataset uses different annotations. Most importantly, it uses a different named entity recognizer, which means that the relation instances differ. Overall, the reparsed system finds fewer instances than the baseline (19998) and performs slightly worse with a precision of 0.56 and a recall of 0.37. A possible explanation for this drop in performance could be that the dependency parses differ, because MINPAR and the Stanford CoreNLP dependency parser use different dependencies. The Stanford CoreNLP parses has more different dependency relation, which possibly creates some confusion for the classifier. Overall the differences are very small, which is also what figure 1 suggests.

The top 5 performing relations mostly overlap with the top 5 of the baseline, although the order is a bit different. The only difference is the number 5, which is in this case /aviation/airport/serves. Even though the actual precision and recall is lower than the number 5 of the baseline, this is a positive change, since the relation is between organisations (airports) and locations (the area it serves). This means that the relation is part of the set of relevant relations, and the performance is therefore more indicative than the /tv/tv_program/genre relation that is in the top 5 of the baseline.

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.69	0.77	0.73	149
/location/location/contains	0.74	0.53	0.62	10009
/geography/river/basin_countries	0.71	0.56	0.63	285
/people/person/place_of_birth	0.62	0.37	0.47	2918
/people/deceased_person/cause_of_death	1.00	0.25	0.40	12

Table 6: Top 5 performing relations of coreference system 1, full set

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.79	0.77	0.78	149
/location/location/contains	0.75	0.53	0.62	10009
/geography/river/basin_countries	0.69	0.57	0.62	285
/people/person/place_of_birth	0.63	0.37	0.47	2919
/people/deceased_person/cause_of_death	1.00	0.25	0.40	12

Table 7: Top 5 performing relations of coreference system 2, full set

Coref1 and Coref2 This section will discuss both coreference systems, each with a different replacement algorithm. The system that replaces all coreferences at once will be referred to as coref1 and the system that only replaces the pair under consideration will be referred to as coref2. The results of these systems are very similar, which is the reason why they are discussed together. The expectation was that both coreference systems would perform better than the reparsed baseline. Considering the overall performance, both systems score a precision of 0.57 and a recall of 0.37, so there is no clear difference between them. The precision is very slightly higher than the reparsed baseline, with 0.01 difference in precision. But figure 1 shows that the performances are barely distinguishable.

Looking at the top 5's of the coreference systems, there is again a large overlap with the reparsed baseline, although the actual numbers tend to be slightly higher. An exception to this is /geography/river/mouth in coref1, which is quite a bit lower than in the reparsed dataset and coref2. A joint difference between the coreference systems is that the relation /people/deceased_person/cause_of_death appears in the top 5 with a precision of 1. For this relation counts the same argumentation as for /tv/tv_program/genre, so the precision and recall number are not a good reflection of the performance of the system.

Even though the overall performance of the different systems seem to be extremely similar, there are some differences outside of the precision and recall values. First of all, both coreference systems find more relation instances than the reparsed baseline (21510 for system 1 and 21511 for system 2), and perform with a similar precision and recall. This means that a relation extraction system that uses coreference information is able to extract more relations from an equal amount of text.

The prediction the replacement algorithm used in coref2 would reduce the feature vocabulary size compared to coref1. This prediction does not hold, since coref1 has a vocabulary size of 3431726, while coref2 has 3555074. The prediction was that coref2 would have a smaller vocabulary, because pronouns will remain intact in the features, where in coref1 these would be replaced with the name the pronoun refers to. A possible reason why this effect doesn't show in the results is the occurence of names in the features. In coref2, these names will always be the names as they appear in the text, so for example the names 'Obama', 'Barrack Obama' and 'Barrack Hussein Obama' can all occur in the features, if they happen to be surrounded by other names that form a relation. On the other hand, in coref1, these names are replaced with their representative mentions, which reduces the variation and results in a smaller vocabulary. The results suggests that this effect is bigger than the effect of pronouns being replaced. The reparsed baseline has a vocabulary of 3408872 features.

5.1.2 Filtered Relations

Figure 3 shows a very similar pattern as figure 1, in the sense that there is no visible difference between the different relation extraction systems. However, it is visible that the precision remains more stable over different recall levels, which means that the filtered set of relations result in a better performing system overall. Appendix B shows the classifications reports for the filtered set of all systems.

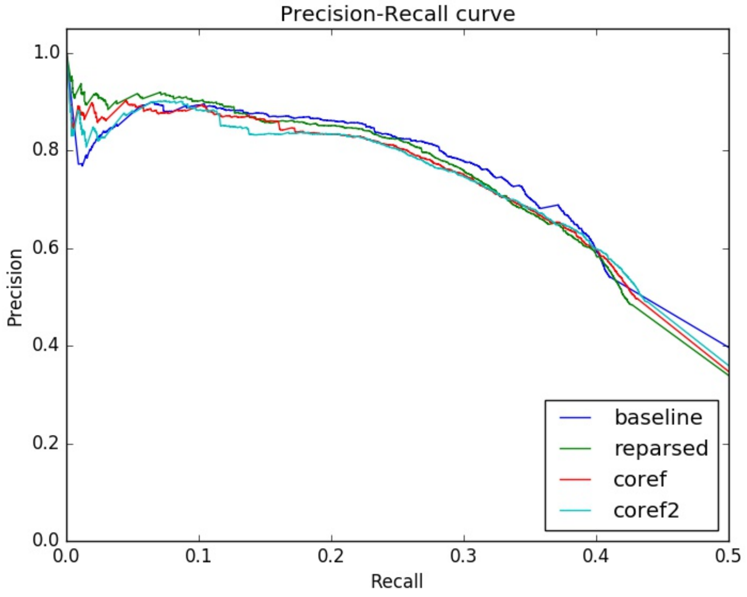


Figure 3: Precision-Recall Curve for the filtered set of relations

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.80	0.80	0.80	149
/people/person/place_of_birth	0.68	0.54	0.60	2299
/location/location/contains	0.80	0.47	0.59	11744
/geography/river/basin_countries	0.72	0.33	0.45	364
/government/political_party/country	0.30	0.17	0.22	105

Table 8: Top 5 performing relations of the baseline, filtered set

Baseline Comparing table 4 and table 8, it is clear that there are many similarities between them. The top 5 performing relations of the filtered baseline don't perform better than the original baseline. The reason why the overall system performs better is because filtering the relations has removed most of the very poorly performing relations. This system has a precision of 0.65 and a recall of 0.38, which is slightly better than the original baseline. Even though the actual increase in precision and recall is quite low, having this filtered set has many advantages. It eliminates many of the badly performing relations that only have a few instances throughout the entire dataset. As a result of this, the system reaches a similar performance, without all these irrelevant classes. This also has a big effect on the feature vocabulary, which is 2340969 for this refiltered baseline. This is more than a million less than the original baseline. Similar to the systems that use the full relation set, there is a wide variety within the number of instances, and again the f-score values drop a lot between the 1st and the last in the top 5. An explanation for this effect can be found in the discussion of the confusion matrix of the filtered set. The number 5 for the baseline is now /government/political_party/country, which has a much lower f-score than the number 5 of the full set baseline, but since this relation generally involves organisations (political parties) and locations (countries), this number is a lot more meaningful than the f-score of /tv/tv_program/genre in the original baseline.

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.74	0.77	0.75	149
/location/location/contains	0.74	0.53	0.62	9987
/geography/river/basin_countries	0.69	0.56	0.62	285
/people/person/place_of_birth	0.59	0.31	0.41	2546
/aviation/airport/serves	0.26	0.36	0.30	143

Table 9: Top 5 performing relations of the reparsed baseline, filtered set

Reparsed The top 5 of the reparsed system is not very different from the top 5 of the full set reparsed system. It includes the same 5 relations and the f-scores are very similar. Just as with the baseline, the overall performance increases a bit with a precision of 0.59 and a recall of 0.39.

Coref1 and Coref2

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.75	0.77	0.76	149
/geography/river/basin_countries	0.75	0.56	0.64	285
/location/location/contains	0.75	0.53	0.62	10009
/people/person/place_of_birth	0.63	0.37	0.46	2918
/government/political_party/country	0.29	0.27	0.28	98

Table 10: Top 5 performing relations of coreference system 1, filtered set

With a precision of 0.60 for both coref systems and a recall of 0.39 for coref1 and a recall of 0.40 for coref2, the performance of the filtered set is again slightly better than the full set. But just as in the full set, the differences with the reparsed dataset are very minimal. Despite the minimal differences in precision and recall, the coreference systems are again able to extract more test instances from the texts, while the overall performance is at least as high as the system without coreference information. This suggests that using coreference information does have an advantage.

relation	precision	recall	f-score	nr. instances
/geography/river/mouth	0.75	0.76	0.75	149
/location/location/contains	0.74	0.53	0.62	10009
/geography/river/basin_countries	0.69	0.56	0.62	285
/people/person/place_of_birth	0.63	0.37	0.47	2919
/government/political_party/country	0.31	0.27	0.29	98

Table 11: Top 5 performing relations of coreference system 2, filtered set

5.1.3 Confusion Matrix

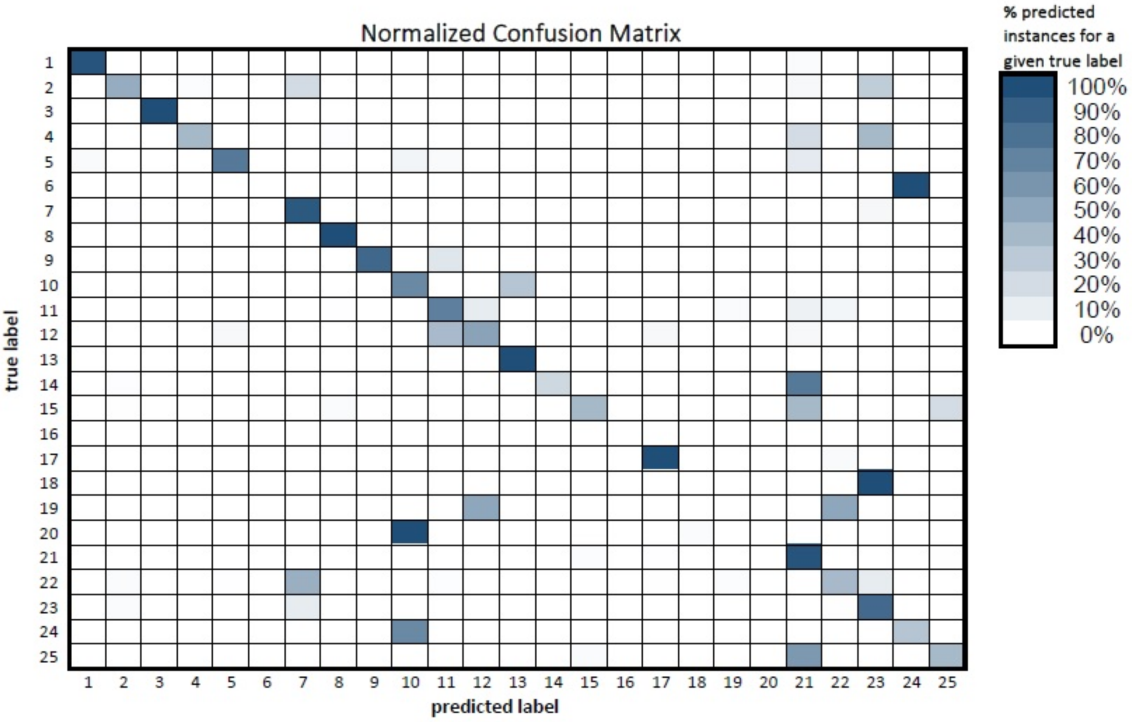


Figure 4: Confusion Matrix for the Reparsed system, filtered set

As mentioned in the earlier sections, some relations perform very well, while the majority of the relations have a fairly low f-score. This can partly be explained by the fact that the system is tuned for high precision and low recall, because of the nature of the feature extraction. This results in behaviour where there needs to be a very good match with the features in order to be able to predict a relation. The majority of the entity pairs in the test set have no relation, and if there is no good match of the test features with any of the relations, the system will typically classify the instance as 'no relation'. Another reason

index	relation
1	/geography/river/basin_countries
2	/people/deceased_person/place_of_death
3	/geography/river/mouth
4	/language/human_language/main_country
5	/aviation/airport/serves
6	/people/person/religion
7	/people/person/place_of_birth
8	/broadcast/radio_station_owner/radio_stations
9	/government/political_party/country
10	/business/company/founders
11	/broadcast/broadcaster/areas_served
12	/broadcast/radio_station/serves_area
13	/fictional_universe/fictional_character/character_created_by
14	/location/us_county/county_seat
15	/geography/mountain_range/mountains
16	/language/human_language/region
17	/people/person/parents
18	/people/person/ethnicity
19	/business/company/place_founded
20	/business/company/major_shareholders
21	/location/location/contains
22	/music/artist/origin
23	/people/person/nationality
24	/architecture/structure/architect
25	/location/country/administrative_divisions

Table 12: indexes of relations in confusion matrix

why some relations fail to be recognised is that some relations are very similar and are sometimes true for many of the same entities. Take for example the relations /people/deceased_person/place_of_death and /people/person/place_of_birth. Many people are born in the same place they eventually die. This means that many instances of the relation /people/deceased_person/place_of_death in freebase will also be instances of /people/person/place_of_birth. Every time these instances occur, their features have increased the weight for both relations. These relations therefore share a lot of features, which makes it harder for the system to make a decision. And since the system only allows one label per instance, people who are born and died in the same place can never be classified 100% correctly.

To illustrate the difficulties of the relation extraction system, consider figure 4. This is the confusion matrix of the reparsed baseline, on the filtered dataset. All confusion matrices look extremely similar, so this one will serve as an example for all systems on the filtered set. Just as in the classification reports, the 'no relation' class is left out of the confusion matrix. So the matrix only shows the instances that are actually classified as one of the relations. This is done to illustrate the precision of the different classes and to show which classes are particularly hard to identify correctly. A good example of a confusion is between /broadcast/radio_station/serves_area and /broadcast/broadcaster/areas_served. These relations are in fact very similar, and even a human would have trouble making the distinction between the two. /broadcast/radio_station/serves_area is limited to radio stations and areas, while /broadcast/broadcaster/areas_served can contain any broadcaster, but also includes radio stations. These relations are largely synonymous and they could probably even be merged. The confusion matrix shows that the relations are so similar that about 50% of the relation /broadcast/radio_station/serves_area are classified as /broadcast/broadcaster/areas_served.

Looking into the most highly weighted features of these two relations, there is some overlap. The most informative feature for these relations is actually identical and represented in table 13. It is a dependency feature with an empty left and right window (which means the feature where both k_left and k_right equals 0), → nsubj station → nmod as dependency middle and ORGANIZATION and LOCATION as named entity tags.

feature	left	NE1	middle	NE2	right
dependency_0_0	[]	O	→ nsubj station → nmod	L	[]

Table 13: highest performing feature for /broadcast/radio_station/serves_area and /broadcast/broadcaster/areas_served

The distinction between these two classes will be typically very hard for any system to make. Not only do the relations share a lot of instances, but there is an actual connection between the two relations. An instance of /broadcast/radio_station/serves_area is per definition an instance of /broadcast/broadcaster/areas_served, since a radio station is a broadcaster, and serves a certain area. This connection is an entailment relation, since one is a subset of the other. The same holds for the relations /music/artist/origin and /people/person/place_of_birth. Instances of the first are quite often classified as the second. Again there is a clear connection between these two relations. The origin of an artist is the place of birth of that person. For the type of relation extraction system described in this thesis, including relations that entail each other will always decrease the performance of the system. Not only because these relations will typically share many features, but also because the system allows only one relation per instance, while in these cases instances of one relation are always an instance of another relation.

Beside the entailment confusions, there are also relations that have many instances in common, or have instances that appear in a similar context, without there being a clear definable relation between them. Two of those confusions are between /business/company/major_shareholders and /business/company/founders and between /people/deceased_person/place_of_death and /people/person/place_of_birth. In the top 5 features for these confusions there is no clear overlap, but it is not hard to imagine how these relations look alike, and will have identical members. These confusions are very intuitive and that the entities having these relations will typically appear in similar contexts. Although for these cases it is not true that the instance of one relation is automatically an instance of another, which means there is no entailment between the relations.

5.1.4 Informative Features

This section will discuss the top informative features of the reparsed, filtered baseline system, highlighting some of the properties of the features of high performing and poorly performing relations. Similar to the confusion matrix, the reparsed baseline is taken as an example, because the different systems have very similar informative features. Table 14 shows the most informative features of the best performing relation throughout all the systems, which is /geography/river/mouth. A thing that stands out is that three of the features are specific and general at the same time. They are specific, because they contain information that clearly refers to rivers, such as the word 'tributary', but it is general in the sense that none of the top features contains words or names that refer to a specific river. This is a very ideal for the relation extraction system, since it clearly defines the context in which this relation will typically occur, while at the same time it doesn't have a bias towards a specific instance in the training set. Many of the features of the highly accurate relations have this property and are therefore quite capable to capture a wide variety of instances. The remaining two features of this example are very general, and it is for a human not possible to see how those are an indication for any specific relation.
The features of classes that don't perform well often have features that are too specific, such as table 15. Many features of this relations show very specific context and contain many details about specific languages, which

feature	inverse	left	NE1	middle	NE2	right
lexical_0	False	[]	L	is/VERB a/CLOSED tributary/NOUN of/CLOSED the/CLOSED	L	[]
dependency_0_0		[]	L	middle:→ nsubj tributary → nmod	L	[]
dependency_1_0		← det the	L	→ conj	L	[]
lexical_0	False	[]	L	,/CLOSED a/CLOSED tributary/NOUN of/CLOSED the/CLOSED	L	[]
lexical_0	True	[]	L	and/CLOSED the/CLOSED	L	[]

Table 14: highest performing features for /geography/river/mouth

feature	inverse	left	NE1	middle	NE2	right
lex1	True	central	L	,/CLOSED and/CLOSED the/CLOSED	O	monastery
lex2	True	to central	L	,/CLOSED and/CLOSED the/CLOSED	O	monastery in
lex0	False	[]	M	language/NOUN ,/CLOSED or/CLOSED kimeru/NOUN ,/CLOSED is/VERB spoken/VERB by/CLOSED the/CLOSED meru/NOUN people/NOUN or/CLOSED the/CLOSED ameru/NOUN who/CLOSED live/VERB on/CLOSED the/CLOSED slopes/NOUN of/CLOSED mount kenya/NOUN ,/CLOSED kenya/NOUN ,/CLOSED	L	[]
lex0	False	[]	O	-rrb-/CLOSED ,/CLOSED nasa/NOUN has/VERB selected/VERB four/NUMBER tal/NOUN sites/NOUN in/CLOSED spain/NOUN and/CLOSED	L	[]
lex0	False	[]	P	's/CLOSED vocabulary/NOUN is/VERB of/CLOSED african/NOUN origin/NOUN ,/CLOSED the/CLOSED most/ADJECTIVE of/CLOSED any/CLOSED creole/NOUN in/CLOSED the/CLOSED	L	[]

Table 15: highest performing features for /language/human_language/region

makes them unsuitable for recognition of instances that didn't occur in the training set.

In order for the system to correctly identify any relation, it needs to be able to extract features that are relation specific, but general enough to capture many different instances. Whenever the features are too specific, or too general, the results are very poor. A relation where this is the case is /people/person/ethnicity (table 16). The first four features are very general, but don't seem to capture any relation specific information, while the fifth feature is so specific it probably only matches the training instance.

feature	inverse	left	NE1	middle	NE2	right
lex0	False	[]	P	is/VERB an/CLOSED	M	[]
lex0	True	[]	M	author/NOUN	P	[]
lex0	False	[]	P	,/CLOSED an/CLOSED	M	[]
dep0_0		[]	M	→amod	P	[]
lex0	False	[]	P	's/CLOSED book/NOUN about/CLOSED a/CLOSED book/NOUN dealer/NOUN who/CLOSED specialises/VERB in/CLOSED	M	[]

Table 16: highest performing features for /people/person/ethnicity

Whether the system is able to extract the right feature set for a relation is of course dependent on the number of instances. But the relation /geography/river/mouth does not have an extremely large amount of instances, and yet the set of features has the right format for the system to be able to classify this correctly. It seems like the inherent properties of the relations and their instances play an important role in the success of the classifier. Some relations will typically have many instances that appear in extremely similar contexts, while others can appear in any kind of text, in any kind of sentence. Which makes it a lot harder to find regularities.

Interesting to note is that the window features for both dependency and lexical features rarely end up in the top 5 most informative features for any relation. This is an indication that just the 'middle' part of each feature is often enough to classify the instance. It could mean that the window features are actually redundant, and the complex conjuncted features are actually only increasing the vocabulary size unnecessarily. Something that would be worth investigating is to split up the complex features into their individual parts, to see how the classifier performs with a smaller vocabulary size. The middle parts themselves are already quite complex and might capture the high precision behaviour that Mintz claims to be preferable and the reason behind the feature conjunction of the baseline.

Also, the way the features are designed might be less than ideal for identifying the 'no relation' class, since it tries to find regularities in the contexts the instances appear. But unrelated entities could occur in every possible context. Entities often appearing in the same context is actually an indication that there is a certain relation between them, and trying to classify unrelated entities in that way does not seem logical. The relation extraction systems in this thesis are quite able to identify unrelated entities, but more because they form the majority of all instances in both training and testing, than that the top features for this class actually describe their unrelatedness.

feature	inverse	left	NE1	middle	NE2	right
lex1	True	,	L	,/CLOSED	L	,
lex1	True	,	L	,/CLOSED	L	and
dep0_0		[]	P	→conj	P	[]
lex0	True	[]	P	was/VERB born/VERB in/CLOSED	L	[]
dep0_0		[]	O	→conj	O	[]

Table 17: highest performing features for NO RELATION

The features of table 17 shows four features that are very general, and one feature that clearly indicates the relation people/person/place_of_birth, even though these are supposed to be the top features for the 'no relation' class. It seems like the syntactic and lexical context is actually a quite bad predictor of the negative class and picks up on false negatives. A way this could potentially be solved is to introduce some other features that are less dependent on the context and could serve as a good predictor of the negative class. For example the distance between the two entities in the sentences, because it seems logical that the further the entities are apart, the more likely it is that they are unrelated. This way the classifier will actually learn 'no relation' specific features and will potentially be able to classify this class correctly based on other factors than it just being the most frequent class in the training data. A possible effect of this is that the precision of the 'no relation' class will be increased, reducing the number of false negatives, and increasing the number correctly classified instance as a relation. A side effect could be that the system loses some of its high precision, low recall nature. Since this feature would be very general, it would match many different cases.

5.1.5 Gold Standard Evaluation

As mentioned before, the Freebase relations of the test instances serve as a gold standard in this automatic evaluation. In this thesis as well as in Mintz (2009), the assumption is that this gold label will generally be correct, even though it is important to be aware that this may not always be the case. In order to get an impression of the correctness of this assumption, 50 sentences are sampled from the best performing relation (/geography/river/mouth), a very poorly performing relation (/language/human_language/main_country), and the 'no relation' class. The samples are taken from the reparsed baseline system for the filtered relation set. For every sample, the number of correctly assigned gold labels is counted.[6]

The correctness of the gold labels in this evaluation is very closely related to the assumption that if two related entities appear in the same sentence together, the relation between them will be expressed in that sentence. By evaluating the quality of the gold standard, this section also evaluates to what extend the assumption holds. This assumption is known to be a problematic aspect of the approach described in Mintz (2009), since it doesn't take into account that an entity pair can have multiple relations. The papers of Riedel (2010), Hoffman (2011) and Surdeanu (2012) propose possible solutions to this.

In the best performing relation (/geography/river/mouth), 43 out of 50

[6] the complete evaluation can be found on https://drive.google.com/a/student.gu.se/folderview?id=0B6ggTdHscw_yb0hDdkw0RWNPNGs&usp=sharing

gold labels are correct. Most sentences have a form similar to the example below.

(8) The Vederoasa River is a tributary of the Danube in Romania.
entity pair: <Vederoasa River, Danube>
gold label: /geography/river/mouth

The cases where the gold label was not correctly assigned were mostly when the two entities appeared in a big list, where there was no relation expressed between the different entities, or when the entities appeared in the same sentence without there being a clear indication of the relation, like in the example below.

(9) No connected resistance was offered to the converging march of Prince Charles's army along the Danube, Khevenh ller from Salzburg towards southern Bavaria, and Prince Lobkowitz from Bohemia towards the Naab.
entity pair: <Naab, Danube>
gold label: /geography/river/mouth

In the very poorly performing relation /language/human_language/main_country (f-score of 0 on the reparsed baseline of the filtered relation set), only 18 out of 50 gold labels are correctly assigned. Evaluating the gold standard for this relation is not straightforward, since it is often hard to determine whether a certain country is the main country in which a language is spoken or if it is one of multiple countries. The choice made in this thesis is to count the gold label as correct, if there is a clear indication in the sentence that a certain language is spoken in a certain country.

The reason for the poor assignment of gold labels for this relation seems to lie in the fact that many language names are also the name of the specific group of people speaking that language. This confusion makes it impossible for the system to identify sentences expressing this relation between languages and locations. Sentences similar to the one below are often incorrectly assigned the relation /language/human_language/main_country.

(10) The Kubu are a tribe inhabiting the foothills of Bukit Barisan in central Sumatra, Indonesia, in the vicinity of the cities of Palembang and Jambi.
entity pair: <Kubu, Indonesia>
gold label: /language/human_language/main_country

The reasoning behind the 'no relation' class is different from the other classes. The assumption here is that all entity pairs that do not form an instance in Freebase, don't express any of the relations. It is unavoidable that this approach results in false negatives, but in order to give an impression of the correctness of this assumption, 50 sentences of the 'no relation' class are sampled. Out of these 50 sentences, 49 were correctly labelled for the 'no relation' class. This means that in those 49 instances, the entity pairs didn't have any of the relations present in the filtered relation set. This does not mean that the entities are entirely unrelated, it only means that the entity pair cannot be assigned any of the relations of the filtered relation set. The example below shows a sentence and an entity pair that are correctly labelled as 'no relation'. In this specific case, the entities are in fact related, but in order to form an instance of the /people/deceased_person/place_of_death relation, the instance should be <person, location> and not <location, person>, which is the case in this specific example.

(11) Michael McEachern McDowell (born 1 June 1950 in Enterprise, Alabama, died 27 December 1999, Boston, Massachusetts) was an American novelist and screenwriter.
entity pair: <Massachusetts, Michael McEachern McDowell>
gold label: no relation

The one sentence from this sample that does form a relation instance is shown in the example below. This is a clear example of a false negative and the only reason why this example has not been recognised is because the entity pair apparently does not form an instance in the test set of Freebase.

(12) He married Catherine Grandison, daughter of William de Grandison, 1st Baron Grandison.
entity pair: <Baron Grandison, Catherine Grandison>
gold label: no relation

The quality of the gold standard is not only important for the correctness of the automated evaluation, but is also essential for training the classifier. This small evaluation shows that the quality of the gold standard varies a lot between the different relations, and there seems to be a strong connection between the performance of the relation and the correctness of the gold labels. In the highest performing relation as well as the 'no relation' class, the gold labels are generally correct and in the very poorly performing relation /language/human_language/main_country the gold label is very often not correct. The fact that for some relations the gold standard is often not correct, is a strong indication that this is a problematic aspect of the Mintz (2009) approach. This small evaluation confirms that the contributions of Riedel (2010), Hoffman (2011) and Surdeanu (2012) are very useful changes to the approach described in Mintz (2009).

5.1.6 Automatic Evaluation: Analysis of Results

The overall picture of the results show that the systems do perform reasonably well as relation extractors. Especially the confusion matrix shows that the classifiers are quite precise on a wide variety of different features. The results also support the argumentation for creating a smaller set of features, that is constructed with the nature of the named entity recognizer in mind.

From this automated evaluation, it is very hard to see what the effect of coreference resolution is, and if this is a positive effect. The difference between the two coreference systems is quite minimal, whether it is on the full relation set or the filtered. The coref1 system has a smaller vocabulary, so computationally speaking this has an advantage over coref2. The difference between the coreference systems and the baseline is unclear. Purely looking at precision and recall values, the differences are extremely small. However, that doesn't mean that there is no advantage in using coreference information at all. First of all, the systems using coreference information are able to extract more relations, with roughly the same precision and recall. Whether this positive result weighs up against the effort the extra layer of annotation requires, is a matter of judgement, but the result in itself is positive and shows that including coreference information increases the amount of information that can be extracted from the text. Second, using coreference information is a way of increasing the amount of data. The reasoning behind the expectation that coreference would improve the results, is that coreference results in more data, which results in more training instances, which results in a better performance. In this prediction it is not taken into account that the Mintz baseline approach may already have a certain ceiling effect. The classifier is trained on 800000 documents, which is such an extreme quantity, that an extension that creates more data possibly doesn't result in a visible improvement. It is possible that with this amount of data, the classifier has reached the maximum of it's performance and adding more data will not make a difference. Therefore it is possible that the prediction about coreference resolution still holds, if the training set were smaller and adding data would have a visible effect. In that case the coreference systems should be able to reach the same result as the baseline, but with less training documents, because the system extracts more instances from an equal amount of documents. If this effect is present, it can be made visible by plotting a learning curve for every system.

The learning curve of the filtered relation set is plotted in figure 5. Each system is trained on 8 training sets, increasing the number of documents with 100000 at a time. If the baseline in fact has reached a ceiling at 800000 documents, the curve should be very flat at that training size, which is exactly what figure 5 shows. The curve is actually so flat that it is barely

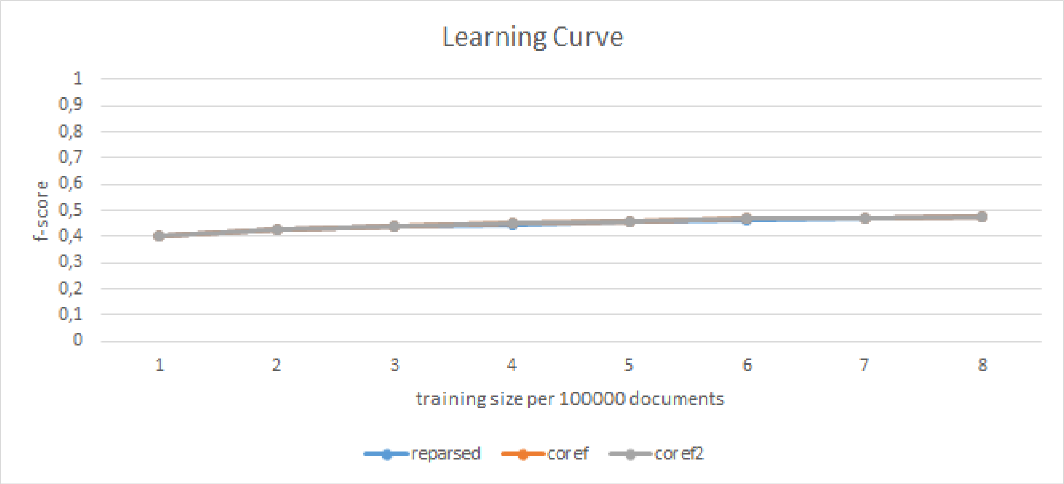


Figure 5: Learning curve for the filtered relation set

a curve. This indicates that there is a very strong ceiling effect, and the figure shows that with 200000 documents, the systems have already almost reached the maximum performance. The plot contains the curve for all systems, but because the ceiling effect is equally present in all systems, the lines follow each other closely, which makes it impossible to distinguish them as separate lines.

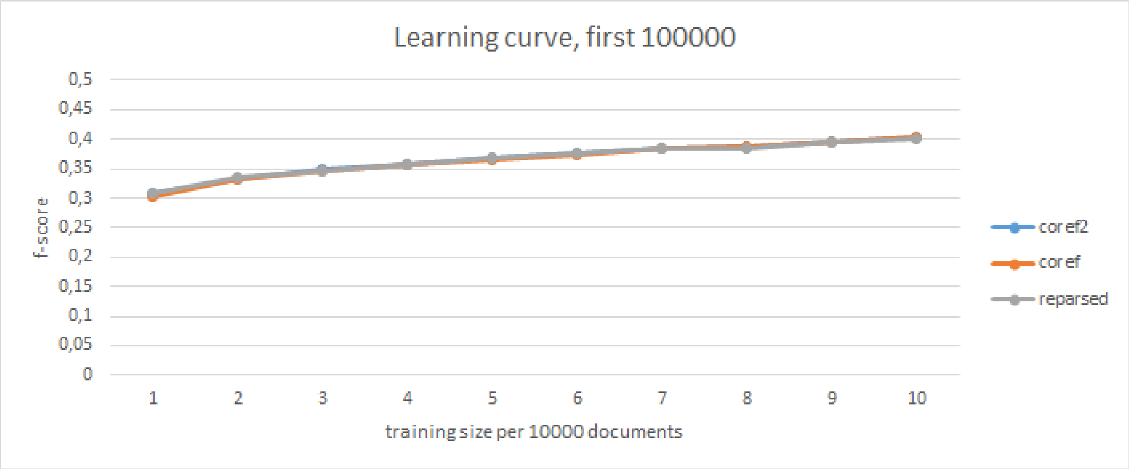


Figure 6: Learning curve for the filtered relation set, first 100000 documents

Because the maximum is reached so early on, it is still not possible to see any effect of coreference resolution. Therefore it is necessary to make a more fine grained learning curve. Figure 6 shows the curve from 0 to 100000 documents, with steps of 10000 at a time. Still the curve is very flat. The classifiers need a lot of extra data for the performance to increase, and still there is no effect of coreference. The plot shows that the training size needs to be increased with a very large amount of documents, to achieve an increase in performance in the f-score. Since the coreference information only adds relatively little extra data, this does not result in an increased f-score.

5.2 Manual Evaluation

In order to be able to give a detailed description of the performance of the system, and to be able to illustrate the relations that are found by the system and where it has trouble, all reparsed classifiers are run on two selected documents. Because the focus of this thesis is the effect of coreference resolution, and these systems can only be fairly compared to the reparsed baseline, only the reparsed systems will be discussed. The documents are neither in the training or test set and are selected for having a relatively short document length, and a clear topic with some relations that are easy to recognise.

Each document has a set of expected relations, based on the set of relations of the respective freebase database. Then the different outputs of the systems are compared to these expected relations to see where the classifiers perform well and where they fail.

The quality of the named entity recognition and the coreference resolution will also be discussed, to give an insight in the dependence of the relation extraction system on these tools, and how errors can work through in the entire system.

The first Wikipedia document that will be discussed is the following:

> (Robert) Paul Robert Elliot (b. 19 December 1931) is an Australian politician. Elliot was born in Ballina, New South Wales and earned a Bachelor of Arts and a MLitt at the University of New England. He worked as a lecturer at the University of Western Sydney, a public servant and electoral officer to John Brown. He was elected as an alderman on Parramatta City Council in 1977 and was mayor from 1984 to 1986. Following the retirement of John Brown, Elliot was elected as the Australian Labor Party member for division of Parramatta in the Australian House of Representatives at the 1990 election. He was appointed Parliamentary Secretary to the Treasurer in the Keating government in December 1993. In June 1994, he became Parliamentary Secretary to the Minister for Communications and the Arts and Parliamentary Secretary to the Min-

ister for Tourism as well. He was defeated at the 1996 election.

Wikipedia

Just looking at the text itself, and the relations in freebase, the relations in table 14 could potentially be found.

relation	entities
/people/person/education	Paul Robert Elliot, Bachelor of Arts
/people/person/education	Paul Robert Elliot, MLitt
/people/person/profession	Paul Robert Elliot, politician
/people/person/nationality	Paul Robert Elliot, Australian
/location/location/contains	New South Wales, Ballina
/government/political_party/country	Australian Labor Party, Australia

Table 18: expected relation from first document

Since the named entity recogniser will select the relation candidates, the extraction system is highly dependent on its output. The educations 'Bachelor of Arts' and 'MLitt' are not recognised, neither is the word 'politician'. The relations containing these words can therefore not be found, which leaves 4 remaining relations. The filtered set is created exactly for this reason, and the remaining relations are indeed the ones that are in this set. Even though the output of the full set and filtered set may not be different, the latter sets far more realistic expectations of the relation extraction systems, taking the capabilities of the preprocessing tools into account.

In case of the relation between 'Paul Robert Elliot' and 'Ballina', the expectation is that there will be a difference between the systems with and without coreference resolution, since in the text the name is referred to as 'Elliot'. The output of the coreference resolution does find a cluster with elements referring to 'Paul Robert Elliot', but unfortunately, the representative mention of this cluster is 'Robert -RRB- Paul Robert Elliott (b. 19 December 1931)'. This is the most complete description of the name that can be found in the text, and the coreference resolution is quite successful in this case, but the representative mention as a whole is not recognised as a name by the named entity recognizer. Because the relation extraction system can only replace coreferences of which the representative is recognised as a name, the name Elliot will not be considered identical to 'Robert Paul Elliot'.

This notation of the representative mention is quite common in Wikipedia articles about persons, so this example probably illustrates a very common inability of the system. This problem is unfortunately very hard to solve. A possibility would be to relax the replacement rules slightly, saying that a coreferent will be replaced by a name if this name is part of the representative mention. The problem with this is that it is quite common that there are multiple names in the representative mention, which would result in wrong mappings.

Considering the preprocessing and the architecture of the system, the expectation is that 4 relations will be found, and that in this case there will be no difference between the systems using coreferences and the baseline. Also, since the remaining 4 relations are all part of the filtered relation set, the expectation is that there will be no difference in output between the filtered and the full relation set.

relation	entities
/location/location/contains	New South Wales, Ballina
/people/person/place_of_birth	Elliott , Ballina

Table 19: found relations from first document

The prediction that all systems should perform the same is indeed true. All systems find the same two relations (table 19), which are both in the expected relations. The behaviour of high precision and low recall is well reflected in this output. It only finds 50% of the relations (considering the remaining 4), which reflects the low recall, but it doesn't find any incorrect relations, which shows the high precision of the system.

The second document that will be discussed is the following:

> Edwin Asa Dix, the pen name of Edwin Augustus Dix (June 25 1860, August 24, 1911), was an American author. Dix was born in Newark, New Jersey, to John Edwin and Mary Joy Dix. He attended the Newark Latin School, then Priceton University from which he graduated in 1881 as Latin Salutatorian, and first in his class with highest grade point average awarded to that date (98.5%). While at Princeton he was managing editor of The Lit and was awarded the Boudinot Historical Fellowship and other prizes. In 1884 he graduated from Columbia Law School with highest honors, and subsequently admitted to the bar in New York and New Jersey. Dix toured the world from 1890-92. On August 15, 1895, he married Marion Alden Olcott at Cherry Valley, New York. They had no children, and spent

> much of their married life abroad, wintering in Egypt, Switzerland, and Colorado. He died suddenly in New York City of myocarditis.
>
> Wikipedia

The expected relations, considering the full set of relations and ignoring the preprocessing, are in table 20.

relation	entities
/people/marriage/spouse	John Edwin Dix, Mary Joy Dix
/people/marriage/spouse	Mary Joy Dix, John Edwin Dix
/people/marriage/spouse	Edwin Asa Dix, Marion Alden Olcott
/people/marriage/spouse	Marion Alden Olcott, Edwin Asa Dix
/people/person/education	Edwin Asa Dix, Newark Latin School
/people/person/education	Edwin Asa Dix, Priceton University
/people/person/education	Edwin Asa Dix, Columbia Law School
/people/person/profession	Edwin Asa Dix, author
/people/deceased_person/cause_of_death	Edwin Asa Dix, myocarditis
/people/person/nationality	Edwin Asa Dix, American
/people/person/parents	Edwin Asa Dix, John Edwin Dix
/people/person/parents	Edwin Asa Dix, Mary Joy Dix
/location/location/contains	New Jersey, Newark
/location/location/contains	New York, Cherry Valley
people/person/place_of_birth	Edwin Asa Dix, Newark
/people/deceased_person/place_of_death	Edwin Asa Dix, New York City

Table 20: expected relations from second document

It is not very straightforward to see what the representative name of 'Dix' should be. Looking at the text, it could be either 'Edwin Asa Dix' or 'Edwin Augustus Dix'.

The named entity recogniser determines which entity pairs are relation candidates. Again, there are some words in the expected relations that are not recognised as a named entity. Just as with the previous document, it is quite predictable which words will not be considered a name, since these are the words that don't fall in the categories Person, Location, Organisation or Miscellaneous. In this case these are the words 'author' and 'myocarditis'. This leaves an impressive 14 relation instances. However, the relations /people/marriage/spouse and /people/person/education belong to the category of relations that only have one or two members in the Freebase data and therefore have an extremely low probability of being found. This eliminates 7 instances from the list of expected relations, and leaves 7 relations that are expected to be found.

Some of the relations cannot be found without coreference information, and some instances will look different without coreference information. The name 'Edwin Asa/Augustus Dix' is referred to as 'Dix' or 'he' in most of the text. The instances where he is referred to as 'Dix' can be recognised, since this is a name, only the system won't be able to see that 'Dix' and 'Edwin Asa Dix' are in fact the same person. The cases where the word 'he' is used, will be completely overlooked in the systems without coreference, which means that certain relations won't be found. From the remaining 7 relations, the place of death of 'Edwin Asa Dix' can only be found by a system that uses coreference information.

Since each system gives a slightly different output, the results of every system on this document are presented in table 21 to 26. Every system finds more relations than are actually in the text. Even though there is variation in the relations the systems find, the systems perform similar in the correct relations they find, and the number of mistakes they make. The /people/person/parents relation between 'Edwin Asa Dix' and his parents is found by none of the systems, neither is his nationality. But the place of birth of 'Dix', and the two /location/location/contains relations are correctly identified by all systems. As expected, the place of death of 'Dix' is in fact found by the systems using coreference resolution, while the other systems fail to do so.

relation	entities	correct
/people/person/place_of_birth	Dix , Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Egypt, Colorado	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/person/nationality	John Edwin, New Jersey	no
/people/person/place_of_birth	Cherry Valley, Marion Alden Olcott	no

Table 21: found relations from second document, baseline, full relation set

relation	entities	correct
/people/person/place_of_birth	Dix, Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Egypt, Colorado	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/person/place_of_birth	Cherry Valley, Marion Alden Olcott	no

Table 22: found relations from second document, baseline, filtered relation set

relation	entities	correct
/people/person/place_of_birth	Edwin Asa Dix, Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Egypt, Colorado	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/person/place_of_birth	Cherry Valley, Marion Alden Olcott	no
/people/deceased_person/place_of_death	Edwin Asa Dix, New York City	yes
/people/deceased_person/place_of_death	New York City, Edwin Asa Dix	no

Table 23: found relations from second document, coref1, full relation set

relation	entities	correct
/people/person/place_of_birth	Edwin Asa Dix, Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Egypt, Colorado	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/person/place_of_birth	Newark, Edwin Asa Dix	no
/people/deceased_person/place_of_death	Edwin Asa Dix, New York City	yes
/people/deceased_person/place_of_death	New York City, Edwin Asa Dix	no

Table 24: found relations from second document, coref2, full relation set

relation	entities	correct
/people/person/place_of_birth	Edwin Asa Dix, Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/person/place_of_birth	Cherry Valley, Marion Alden Olcott	no
/people/deceased_person/place_of_death	Edwin Asa Dix, New York City	yes
/people/deceased_person/place_of_death	New York City, Edwin Asa Dix	no

Table 25: found relations from second document, coref1, filtered relation set

relation	entities	correct
/people/person/place_of_birth	Edwin Asa Dix, Newark	yes
/location/location/contains	New Jersey, Newark	yes
/location/location/contains	New York, Cherry Valley	yes
/location/location/contains	Switzerland, Egypt	no
/location/location/contains	Egypt, Colorado	no
/location/location/contains	Colorado, Egypt	no
/people/deceased_person/place_of_death	Marion Alden Olcott, Cherry Valley	no
/people/deceased_person/place_of_death	Cherry Valley, Marion Alden Olcott	no
/people/deceased_person/place_of_death	Edwin Asa Dix, New York City	yes
/people/deceased_person/place_of_death	New York City, Edwin Asa Dix	no

Table 26: found relations from second document, coref2, filtered relation set

5.2.1 Manual Evaluation: Analysis of Results

The relations that are found for both documents, are relations that have a high performance in the automatic evaluation. The relations that are found in this manual evaluation are all expressed in a very unambiguous way, and the entities in the pairs are very close together in the sentences.

The relations /people/person/place_of_birth and /location/location/contains are very often expressed in wikipedia texts, and often in a very similar way. An article about a person typically contains a sentence of the following form:

(13) PERSON was born in LOCATION, LOCATION

The sentence 'Astronomer Edwin Hubble was born in Marshfield, Missouri', which is the example sentence given in Mintz (2009), falls into this pattern. In these sentences it is very clear what the relations are, and because these sentences are very frequent in Wikipedia texts, the features associated with them are very strong indicators for those relations. This straightforward way these relations are expressed is probably the main reason why these specific relations perform so well, and why the systems have problems recognising some other relations. Most relations do not have such a clear re-occurring template.

For the first document that is discussed in the previous section, no incorrect relations are found, but for the second document, which contains many more related entity pairs, this is not the case. For the second document, every system finds 5 or 6 incorrect relations. Some of those relations are actually not existent in the text, but some others are the inverse of one of the correct relations, such as the place of death of Dix. In some systems both the tuples <person,location> and <location,person> are classified for having this relation. This is an interesting observation, because these kind of errors in asymmetrical relations are supposed to be prevented by the boolean value that is part of every lexical feature (see Feature Extraction). This boolean makes sure that the order of the entities is included in the features, to prevent the same relations to hold between tuples and their opposite.

A possible explanation why these opposite entity pairs occur, is because there is no strict constraint on what the members of the relations can be. The idea is that this constraint will develop by itself, by including the boolean on the lexical features, and by simply including the named entity tags and their order in every feature. But since all annotation is done automatically, there will be exceptions in the training data where the entities for example have an incorrect named entity tag, leading to the systems being trained on instances that don't conform to the expected entities for that relation. An alternative explanation why these inverse relations are found by the system, could be that the dependency features don't have the boolean that indicates whether the order of the entities is inverse or not. This means that the constraint of which entity comes first in the sentence is not so strong for dependency features, and there is a possibility that inverse relation tuples match to these features. A possible way these kinds of errors could be prevented is by defining the types of tuples that can be relation instances as a constraint for every relation. For all relations in the filtered set, there can be made a clear definition of what an instance can look like.

Another type of error that is made by the classifiers is overgeneralisation. The most clear example of this is the /location/location/contains relation that is found between Switzerland and Egypt. In the text these words are only separated by a comma, which is exactly the context that is very frequent for locations that do have this relation. In fact, the very frequent pattern of example 7 contains exactly this construction. The features that are extracted from this pattern, whether it is for people/person/place_of_birth or /location/location/contains will be very strongly associated with these relations, because of the frequency of this pattern. The features for two locations separated by a comma are such a strong indicator for the /location/location/contains relation, that even when this relation is not actually present, it is often classified as such.

From the relations that are not recognised by any of the classifiers, /people/person/nationality stands out the most. This relation occurs in both documents, and seems to fall into a pattern that is actually very frequent in Wikipedia articles. The nationality of people is often expressed in a template similar to the following:

(14) PERSON (DATE), is/was an NATIONALITY PROFESSION.

A sentence such as 'Mary (1992-09-05) is a Swedish linguist' falls in to this template. Even though this is probably a very frequent pattern, still the system is unable to adequately identify this relation. The reason for this is the date between the brackets. This slot often contains the date of birth of the person in question. Since this slot appears in the text between the two entity pairs, this will be included in the 'middle' part of the features. This date is of course different for most people, but because features will only match if they are identical, the system is unable to match two features where this date is the only difference. This most likely stands in the way for the system to find a regularity.

There are multiple ways this kind of behaviour could be solved, which would potentially improve the performance of the entire classifier. Mintz constructed the features in such a way that the resulting classifier has a high precision, low recall performance. However, in cases like this the system seems too precise, and it seems preferable if there would be a match in features if there is only a minor difference between two features, such as a date. A possible way to solve this would to replace certain elements in the text with the same token. For example, the Stanford CoreNLP named entity recogniser is able to recognise dates. For this specific case, the problem would be solved by replacing every occurrence of a date with the word 'date'. However, this would mean that in order to optimize the classifier, it is necessary to identify all elements that cause these kinds of problems.

An alternative way would be to introduce some more general features to the system. For example, beside using the entire middle part, it would be possible to add a feature that contains the verbs between the two entities (if present). For example 8, this would result in: PERSON was/is NATIONALITY. This should make it possible for the system to find more regularities, and to correctly recognise patterns like example 8. A side effect of introducing such a feature could be that the system loses some of its low recall, high precision behaviour, since the proposed feature would be very general and would match many different contexts. What the exact effect of these solutions would be, could be a topic of further research.

6 Discussion and Conclusion

The goal of this thesis was to show the effect of coreference information on a relation extraction system, and to give a detailed analysis of the method described in Mintz.

The effect of coreference information on the relation extraction system used in this thesis does not show in the precision and recall values. However, there are certain advantages of using this extra information. First of all, using coreference resolution clusters mentions that refer to the same entity. This makes it possible for the system to see that mentions such as 'Obama' and 'Barack Obama' are in fact the same entity. Another advantage is that the systems using coreference information are able to extract more entities than the baseline, from an equal amount of text, while keeping the same precision and recall scores as the baseline. The fact that there is no visible increase in performance in the automatic evaluation could be due to the fact that in order to increase the performance, the system needs a lot of extra data. The amount of data that coreference resolution adds is not enough to achieve this amount. Another reason why the effect is minimal, is possibly because automated coreference resolution still makes a lot of errors. Not all entities are clustered in the right way, which is a source of noise in the training data.

This thesis also described the exact architecture and performance of the approach described in Mintz (2009). Even though the system performs well, especially considering it's simplicity, there are aspects that could be improved. The set of Freebase relations contains many relations that the named entity recogniser is unable to deal with, which makes it impossible for the classifier to get the right training data for these relations. Using relations that contain the right entity types results in a better performing system.

Another aspect that could be improved is the feature selection. The large conjunctions of features in Mintz (2009) has the effect that the system performs with high precision and low recall, but this thesis shows that this effect is in some cases too strong. Minor differences in the features can lead to the system being unable to find regularities in the way certain relations are expressed.

In the analysis of the baseline, it has become clear that even though distant supervision makes it possible to train a classifier on a very large amount of data, this is not always useful. The training set in Mintz (2009) consists of 800000 documents. By plotting a learning curve for this system, it has become clear that this is in fact an unnecessarily large number. Roughly the same performance can be achieved by just 200000 documents.

The analysis of the approach described in Mintz (2009) shows that roughly the same results can be achieved in a reimplementation. It also addresses some issues, which show that the method can be improved with a stricter selection of the relation set. Some improvements in the feature design are proposed, which could potentially lead to better results, without adding to the complexity of the algorithm.

This thesis has showed that even though there is no increase in precision and recall, using coreference resolution in a relation extraction system has advantages and a lot of potential. The expectation is that the effect will become more visible as the quality of automated annotation tools will increase in the future.

7 Future Work

This thesis suggests some improvements in the feature selection in the approach described in Mintz. These include more general features that would make it possible to find regularities in more relations, and features that specifically address the problem of finding the negative class. It would be interesting to see if these suggestions indeed lead to an improvement and how those weigh up against improvements that would make the algorithm much more complex, such as Riedel (2010), Hoffman (2011), Surdeanu (2012) and Zeng (2015).

In this thesis, the Freebase relations are re-filtered in order to match the behaviour of the named entity recogniser. An alternative approach to this would be to solve this problem at the level of the named entity recogniser. By making sure all entities in the set of relations can be identified by the named entity recogniser, high quality training data can be extracted for every relation. This could potentially improve the results, without reducing the set of relations.

Automated coreference resolution is a topic of research in itself and many improvements can be made. The expectation is that as the quality of coreference resolution increases, its effect on automated relation extraction will become more visible.

A Freebase Relations

A.1 Baseline

```
/american_football/football_position/players
/architecture/structure/architect
/automotive/model_year/body_styles
/automotive/model_year/engines
/automotive/model_year/exterior_colors
/automotive/model_year/make
/automotive/model_year/model
/automotive/model_year/next_model_year
/automotive/model_year/previous_model_year
/automotive/model_year/transmissions
/aviation/aircraft_model/comparable_aircraft
/aviation/aircraft_model/manufacturer
/aviation/airport/serves
/baseball/baseball_position/players
/basketball/basketball_player/position_s
/biology/organism_classification/higher_classification
/biology/organism_classification/rank
/book/author/works_written
/book/book/editions
/book/book/genre
/book/book_edition/author_editor
/book/written_work/original_language
/broadcast/broadcaster/areas_served
/broadcast/content/artist
/broadcast/content/broadcast
/broadcast/genre/content
/broadcast/podcast_feed/publication_frequency
/broadcast/radio_station/serves_area
/broadcast/radio_station_owner/radio_stations
/business/business_chain/location
/business/company/founders
/business/company/headquarters
/business/company/industry
/business/company/major_shareholders
/business/company/place_founded
/business/industrial_classification/parent
/business/industry/parent_industry
/dining/restaurant/cuisine
/education/education/degree
```

```
/education/education/institution
/education/educational_institution/colors
/education/educational_institution/school_type
/fictional_universe/fictional_character/character_created_by
/fictional_universe/fictional_character/gender
/fictional_universe/fictional_character/occupation
/fictional_universe/fictional_character/powers_or_abilities
/fictional_universe/fictional_character/species
/fictional_universe/fictional_universe/characters
/film/actor/film
/film/director/film
/film/film/cinematography
/film/film/country
/film/film/edited_by
/film/film/genre
/film/film/language
/film/film/music
/film/film/sequel
/film/producer/film
/film/writer/film
/geography/mountain/mountain_type
/geography/mountain_range/mountains
/geography/river/basin_countries
/geography/river/mouth
/government/political_party/country
/influence/influence_node/influenced
/language/human_language/main_country
/language/human_language/region
/language/language_family/languages
/library/public_library_system/central_library
/location/country/administrative_divisions
/location/location/contains
/location/location/time_zones
/location/us_county/county_seat
/metropolitan_transit/transit_line/stops
/music/artist/origin
/music/composition/composer
/music/lyricist/lyrics_written
/people/deceased_person/cause_of_death
/people/deceased_person/place_of_death
/people/marriage/spouse
/people/person/education
/people/person/employment_history
```

```
/people/person/ethnicity
/people/person/nationality
/people/person/parents
/people/person/place_of_birth
/people/person/profession
/people/person/religion
/soccer/football_position/players
/tv/tv_program/country_of_origin
/tv/tv_program/episodes
/tv/tv_program/genre
/tv/tv_program/program_creator
/tv/tv_series_episode/director
/tv/tv_series_episode/writer
/tv/tv_series_season/episodes
/user/bio2rdf/public/bm/references
/visual_art/artwork/artist
/visual_art/artwork/media
/visual_art/visual_artist/art_forms
/visual_art/visual_art_form/artworks
/wine/wine/grape_variety
```

A.2 re-filtered

```
/architecture/structure/architect
/aviation/airport/serves
/broadcast/broadcaster/areas_served
/broadcast/radio_station/serves_area
/broadcast/radio_station_owner/radio_stations
/business/company/founders
/business/company/major_shareholders
/business/company/place_founded
/fictional_universe/fictional_character/character_created_by
/geography/mountain_range/mountains
/geography/river/basin_countries
/geography/river/mouth
/government/political_party/country
/language/human_language/main_country
/language/human_language/region
/location/country/administrative_divisions
/location/location/contains
/location/us_county/county_seat
/music/artist/origin
/people/deceased_person/place_of_death
```

```
/people/person/ethnicity
/people/person/nationality
/people/person/parents
/people/person/place_of_birth
/people/person/religion
```

B Classification Reports Re-Filtered Relations

B.1 Baseline

```
relation
precision recall f1-score nr. instances


/architecture/structure/architect
0.07      0.04      0.05        28
/aviation/airport/serves
0.18      0.17      0.17       147
/broadcast/broadcaster/areas_served
0.15      0.15      0.15       174
/broadcast/radio_station/serves_area
0.32      0.10      0.15       122
/broadcast/radio_station_owner/radio_stations
0.24      0.09      0.13        43
/business/company/founders
0.02      0.07      0.03        29
/business/company/major_shareholders
0.00      0.00      0.00         2
/business/company/place_founded
0.00      0.00      0.00        40
/fictional_universe/fictional_character/character_created_by
0.05      0.03      0.03        77
/geography/mountain_range/mountains
0.07      0.04      0.05        51
/geography/river/basin_countries
0.72      0.33      0.45       364
/geography/river/mouth
0.80      0.80      0.80       148
/government/political_party/country
0.30      0.17      0.22       105
/language/human_language/main_country
0.25      0.03      0.06        60
/language/human_language/region
0.00      0.00      0.00        21
/location/country/administrative_divisions
0.11      0.18      0.14        92
/location/location/contains
0.80      0.47      0.59     11744
/location/us_county/county_seat
```

```
0.13      0.15      0.14       124
/music/artist/origin
0.36      0.11      0.17       479
/people/deceased_person/place_of_death
0.15      0.07      0.09       337
/people/person/ethnicity
0.00      0.00      0.00        12
/people/person/nationality
0.36      0.13      0.19      3346
/people/person/parents
0.21      0.13      0.16        98
/people/person/place_of_birth
0.68      0.54      0.60      2299
/people/person/religion
0.00      0.00      0.00        40

avg / total       0.65      0.38      0.47     19982
```

B.2 Reparsed

```
/architecture/structure/architect
0.00      0.00      0.00        22
/aviation/airport/serves
0.26      0.36      0.30       143
/broadcast/broadcaster/areas_served
0.21      0.28      0.24       157
/broadcast/radio_station/serves_area
0.29      0.16      0.21       101
/broadcast/radio_station_owner/radio_stations
0.22      0.05      0.08        42
/business/company/founders
0.04      0.13      0.06        38
/business/company/major_shareholders
0.00      0.00      0.00         3
/business/company/place_founded
0.00      0.00      0.00        42
/fictional_universe/fictional_character/character_created_by
0.18      0.06      0.08        90
/geography/mountain_range/mountains
0.44      0.20      0.28        40
/geography/river/basin_countries
0.69      0.56      0.62       285
```

```
/geography/river/mouth
0.74      0.77      0.75       149
/government/political_party/country
0.31      0.26      0.28        98
/language/human_language/main_country
0.00      0.00      0.00        54
/language/human_language/region
0.00      0.00      0.00        20
/location/country/administrative_divisions
0.16      0.29      0.21        73
/location/location/contains
0.74      0.53      0.62      9987
/location/us_county/county_seat
0.21      0.18      0.19       115
/music/artist/origin
0.46      0.10      0.16       447
/people/deceased_person/place_of_death
0.15      0.08      0.10       345
/people/person/ethnicity
0.00      0.00      0.00        14
/people/person/nationality
0.37      0.15      0.21      3687
/people/person/parents
0.28      0.18      0.22       101
/people/person/place_of_birth
0.59      0.31      0.41      2546
/people/person/religion
0.00      0.00      0.00        59


avg / total       0.59      0.39      0.46     18658
```

B.3 Coref1

```
/architecture/structure/architect
0.00      0.00      0.00        25
/aviation/airport/serves
0.21      0.28      0.24       137
/broadcast/broadcaster/areas_served
0.19      0.28      0.23       155
/broadcast/radio_station/serves_area
0.40      0.16      0.23        99
/broadcast/radio_station_owner/radio_stations
```

```
0.15      0.05      0.07        42
/business/company/founders
0.04      0.17      0.07        42
/business/company/major_shareholders
0.00      0.00      0.00         3
/business/company/place_founded
0.00      0.00      0.00        42
/fictional_universe/fictional_character/character_created_by
0.20      0.09      0.12       105
/geography/mountain_range/mountains
0.54      0.17      0.25        42
/geography/river/basin_countries
0.75      0.56      0.64       285
/geography/river/mouth
0.75      0.77      0.76       149
/government/political_party/country
0.29      0.27      0.28        98
/language/human_language/main_country
0.00      0.00      0.00        55
/language/human_language/region
0.00      0.00      0.00        20
/location/country/administrative_divisions
0.13      0.25      0.17        76
/location/location/contains
0.75      0.53      0.62     10009
/location/us_county/county_seat
0.25      0.18      0.21       119
/music/artist/origin
0.44      0.09      0.15       452
/people/deceased_person/place_of_death
0.24      0.14      0.18       453
/people/person/ethnicity
0.00      0.00      0.00        15
/people/person/nationality
0.42      0.18      0.26      4607
/people/person/parents
0.18      0.18      0.18        94
/people/person/place_of_birth
0.63      0.37      0.46      2918
/people/person/religion
0.00      0.00      0.00        72
```

```
avg / total       0.60      0.39      0.47     20114
```

B.4 Coref2

```
/architecture/structure/architect
0.00      0.00      0.00         25
/aviation/airport/serves
0.21      0.29      0.24        137
/broadcast/broadcaster/areas_served
0.21      0.30      0.24        155
/broadcast/radio_station/serves_area
0.29      0.16      0.21         99
/broadcast/radio_station_owner/radio_stations
0.11      0.05      0.07         42
/business/company/founders
0.06      0.14      0.08         42
/business/company/major_shareholders
0.00      0.00      0.00          3
/business/company/place_founded
0.00      0.00      0.00         42
/fictional_universe/fictional_character/character_created_by
0.20      0.10      0.13        105
/geography/mountain_range/mountains
0.32      0.17      0.22         42
/geography/river/basin_countries
0.69      0.56      0.62        285
/geography/river/mouth
0.75      0.76      0.75        149
/government/political_party/country
0.31      0.27      0.29         98
/language/human_language/main_country
0.00      0.00      0.00         55
/language/human_language/region
0.00      0.00      0.00         20
/location/country/administrative_divisions
0.15      0.26      0.19         76
/location/location/contains
0.74      0.53      0.62      10009
/location/us_county/county_seat
0.27      0.18      0.21        119
/music/artist/origin
0.47      0.09      0.16        452
/people/deceased_person/place_of_death
```

```
0.24       0.17       0.20        453
/people/person/ethnicity
0.00       0.00       0.00         15
/people/person/nationality
0.42       0.20       0.27       4607
/people/person/parents
0.16       0.19       0.18         94
/people/person/place_of_birth
0.63       0.37       0.47       2919
/people/person/religion
0.00       0.00       0.00         72


avg / total       0.60       0.40       0.47      20115
```

References

ACE. (2000-2005). Automatic content extraction. Retrieved from `https://www.ldc.upenn.edu/collaborations/past-projects/ace`

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction for the web. In Ijcai (Vol. 7, pp. 2670–2676).

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In Sigmod '08: Proceedings of the 2008 acm sigmod international conference on management of data (pp. 1247–1250). New York, NY, USA: ACM.

Bunescu, R., & Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. In Annual meeting-association for computational linguistics (Vol. 45, pp. 576–583).

Chan, Y., & Roth, D. (2010). Exploiting background knowledge for relation extraction. In Coling. Beijing, China.

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In Emnlp (pp. 740–750).

Clark, K., & Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In Association of computational linguistics (acl).

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 363–370).

Florian, R., Hassan, H., Ittycheriah, A., Jing, H., Kambhatla, N., Luo, X., … Roukos, S. (2004). A statistical model for multilingual entity detection and tracking (Tech. Rep.). New York,US: DTIC Document.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., & Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - volume 1 (pp. 541–550). Stroudsburg, PA, USA: Association for Computational Linguistics.

Ittycheriah, A., Lita, L., Kambhatla, N., Nicolov, N., Roukos, S., & Stys, M. (2003). Identifying and tracking entity mentions in a maximum entropy framework. In 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 (p. 40-42).

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In Proceedings of the acl 2004 on interactive poster and demonstration sessions (p. 22-2).

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th international conference on machine learning (p. 282-289). San Fransisco, United States.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., & Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics, 39(4), 885–916.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., & Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In Proceedings of the fifteenth conference on computational natural language learning: Shared task (pp. 28–34).

Lin, D. (2003). Dependency-based evaluation of minipar. In A. Abeillé (Ed.), Treebanks (Vol. 20, p. 317-329). Springer Netherlands.

Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., & Roukos, S. (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In Proceedings of the 42nd annual meeting on association for computational linguistics (p. 135-142).

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Association for computational linguistics (acl) system demonstrations (pp. 55–60). Retrieved from http://www.aclweb.org/anthology/P/P14/P14-5010

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. Computational linguistics, 19(2), 313–330.

Metaweb. (2008). Freebase data dumps. Retrieved from http://download.freebase.com/datadumps/

Miller, G. A. (1995). Wordnet: a lexical database for english. Communications of the ACM, 38(11), 39–41.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In Proceedings of the 47th annual meeting of the association for computational linguistics (p. 1003-1011).

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In Proceedings of the workshop on incremental parsing: Bringing engineering and cognition together (pp. 50–57).

Nivre, J., Agić, Ž., Aranzabe, M. J., Asahara, M., Atutxa, A., Ballesteros, M., … others (2015). Universal dependencies 1.2.

Nivre, J., Hall, J., & Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In Proceedings of lrec (Vol. 6, pp. 2216–2219).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., & Manning, C. (2010). A multi-pass sieve for coreference resolution. In Proceedings of the 2010 conference on empirical methods in natural language processing (pp. 492–501).

Recasens, M., de Marneffe, M.-C., & Potts, C. (2013). The life and death of discourse entities: Identifying singleton mentions. In Hlt-naacl (pp. 627–633).

Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In Proceedings of the 2010 european conference on machine learning and knowledge discovery in databases: Part iii (pp. 148–163). Berlin, Heidelberg: Springer-Verlag.

Shinyama, Y., & Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics (pp. 304–311).

Snow, R., Jurafsky, D., & Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), Advances in neural information processing systems 17 (pp. 1297–1304). Cambridge, MA: MIT Press.

Surdeanu, M., Tibshirani, J., Nallapati, R., & Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (pp. 455–465). Stroudsburg, PA, USA: Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1 (pp. 173–180).

Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, & Y. Marton (Eds.), Emnlp (p. 1753-1762). The Association for Computational Linguistics.

Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005). Exploring various knowledge in relation extraction. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 427–434). Stroudsburg, PA, USA: Association for Computational Linguistics.