Chapter 6

Using lexico-semantic knowledge for question answering

Part of this chapter is taken from Van der Plas et al. (2008a), Bouma et al. (2007), and Mur and Van der Plas (2007).

6.1 Introduction

In the previous chapters we have seen three methods for acquiring lexico-semantic information: the syntax-based method, the alignment-based method, and the proximity-based method. We have seen that the three methods give rise to nearest neighbours that are very different in nature.

We evaluated the nearest neighbours on two gold standards: Dutch EuroWordNet (Vossen, 1998) and the Dutch association norms (De Deyne and Storms, 2008).

The syntax-based method results in nearest neighbours that are semantically related. They belong to the same semantic category. Many of the nearest neighbours are co-hyponyms. Even at the first ranks, about twice as many co-hyponyms are found than synonyms. The number of hypernyms and hyponyms depends on the frequency of the test word, but is usually a little bit smaller than the number of synonyms.

The alignment-based method gives rise to many synonyms. The ratio between the alignment-based method and the syntax-based method with respect to percentages of synonyms found is approximately 3:2.

The proximity-based method finds fewer semantically related words, but
Chapter 6. Using lexico-semantic knowledge for question answering

more associations than the two other methods. The ratio between the proximity-based method and the syntax-based method with respect to percentages of associations found is approximately 2:1, if the same corpora are used. The ratio between the proximity-based method and the alignment-based method is around 3:2.

In this chapter we would like to evaluate the retrieved nearest neighbours on a task: open-domain question answering (QA). In open-domain QA the system returns brief answer strings to questions posed in natural language by (pseudo) users. The open-domain QA system Joost developed in Groningen has several components, which we will describe briefly in the next section (6.2). We would like to find out if and for what modules the acquired lexico-semantic information is useful. We will describe what types of information we have used in QA in section 6.3. Sections 6.4 up to 6.7 discuss the application of lexico-semantic information to each of the components. Finally we will conclude with reflections on the usefulness of lexico-semantic knowledge for QA.

6.2 The architecture of Joost

In Figure 6.1 we can see the architecture of Joost sketched. Besides the three classical processing stages QUESTION ANALYSIS, PASSAGE RETRIEVAL, and ANSWER EXTRACTION, the system also contains a component that is based on the technique of extracting answers off-line: OFF-LINE ANSWER EXTRACTION and TABLE LOOK-UP. All components in our system rely heavily on syntactic analysis, which is provided by Alpino (Van Noord, 2006), a wide-coverage dependency parser for Dutch. We parsed both question and document collection with Alpino. We will now give a brief overview of the components in our QA system. The components will be explained in more detail in sections 6.4 up to 6.7, where the application of lexico-semantic information to each component will be discussed.

The first processing stage is QUESTION ANALYSIS. The input to this component is a natural language question in Dutch, which is parsed by Alpino. The task of question analysis is to determine the question type and to identify keywords in the question. Questions are classified according to the expected answer type. A question like What country is the biggest producer of vodka? would be classified as a LOCATION question because the expected answer type is a named entity of the type LOCATION.

From question analysis we can take two directions. Depending on the question type the next stage is either INFORMATION RETRIEVAL (IR) or TABLE LOOK-UP. If the question is classified as a question for which tables exist, it will be answered by table look-up. Answers to highly likely questions, for which
fixed patterns can be defined, are stored in tables before the question answering process takes place. Facts are extracted from the parsed text collection using these fixed patterns. Potential answers together with the IDs of the paragraphs in which they were found are stored. During the question answering process the question type determines which table is selected (if any) and the keywords help to find and rank the paragraphs that might contain the correct answer. We apply some ranking heuristics to the off-line component as well. We do this to make the chances of selecting the correct answer more likely.

For all questions that cannot be answered by table look-up, we follow the other path through the QA system to the passage retrieval component. Previous experiments have shown that a segmentation of the corpus into paragraphs is most efficient for information retrieval (IR) performance in QA. Hence, IR passes relevant paragraphs to subsequent modules for extracting the actual answers from these text passages.

The final processing stage in our QA system is answer extraction and selection. The input to this component is a set of paragraph IDs, either provided by off-line QA or by the IR system. We then retrieve all sentences
from the text collection included in these paragraphs. For questions that are answered by means of table look-up, the tables provide an exact answer string. In this case the context is used for ranking the answers. For other questions, it is necessary to extract answer strings from the paragraphs returned by IR. Several features are used to rank the extracted answers. Finally, the answer ranked first is returned to the user.

### 6.3 Lexico-semantic information used

The lexico-semantic information used in this chapter for the various QA components includes, but is not limited to, the three types we have seen in the previous chapters.

- Nearest neighbours from syntax-based distributional similarity
- Nearest neighbours from alignment-based distributional similarity
- Nearest neighbours from proximity-based distributional similarity

We gathered nearest neighbours for a frequency-controlled list of words that was still manageable to retrieve. We included all words (nouns, verbs, adjectives and proper names) with a frequency of 150 and higher in the CLEF corpus (80M words). It is the same corpus we also use for the QA experiments. This resulted in a ranked list of nearest neighbours for the 2,387 most frequent adjectives, the 5,437 most frequent nouns, the 1,898 most frequent verbs, and the 1,399 most frequent proper names. For all words and for each type of similarity we retrieved a ranked list of its 100 nearest neighbours with accompanying similarity score.\(^1\)

In Table 6.1 in the first three rows we see the amount of information that is contained in individual distributional lexico-semantic resources. It is clear from the numbers that the alignment-based method does not provide nearest neighbours for all headwords selected. Only 4.0K nouns from the 5.4K retrieve

\(^{1}\)Note that we will not use the full list of 100 nearest neighbours for all experiments.
6.3. Lexico-semantic information used

nearest neighbours. The data is sparse. Also, the alignment-based method
does not have any nearest neighbours for proper names, due to decisions we
made earlier regarding preprocessing: All words were transformed to lowercase.
The proximity-based method also misses a number of words, but the number is
far less important. The amount of information the lists of categorised named
eentities provide is much larger than the amount of information comprised in the
list provided by distributional methods.

In the last three rows of Table 6.1 we see three additional types of lexico-
semantic information we used. In addition to the lexico-semantic information
resulting from the three distributional methods we used:

- Dutch EuroWordNet (Vossen, 1998)
- Categorised named entities

With respect to the first resource we can be brief. We selected the synsets
of this lexico-semantic resource for nouns, verbs, adjectives and proper names.
Numbers are given in Table 6.1.²

The categorised named entities are a by-product of the syntax-based distri-
butional method. From the example in (1) we extract the apposition relation
between Van Gogh and schilder ‘painter’.

(1)  Van Gogh, de beroemde schilder huurde een atelier, Het Gele huis, in
Arles.
‘Van Gogh, the famous painter, rented a studio, The Yellow House, in
Arles.’

The apposition relation was used to compare words in a distributional frame-
work in Chapter 3. We used the apposition relation to determine the similarity
between two words, such as Van Gogh and Rembrandt. However, the fact that
Van Gogh is an instance of the category of painters is valuable information in
itself that can be very useful for several components of our QA system. Whereas
we used the apposition relation to determine the distributional similarity, the
second-order affinity between words, we now use the first-order affinities be-
tween a named entity and a category directly: the instance Van Gogh belongs
to the category of painters. There is an instance relation between the named
entity Van Gogh and painter, i.e. Van Gogh is a painter.

Named entities are typically not very well represented in existing resources
such as WordNet. As Pasca and Harabagiu (2001) explain regarding Princeton
WordNet, “the hyponyms of concepts such as composer or poet are illustrations

²Note that the number of nouns from EWN is the result of subtracting the proper names.
rather than an exhaustive list of instances. For example, only twelve composer
names specialize concept composer”.

Apart from applying the apposition relation to acquire categorised named
entities, for which we gave results in Van der Plas and Bouma (2005b), we used
the relation of nominal predicate complement.\(^3\) This pattern-based approach
is very much related to work done by Hearst (1992) and by IJzereef (2005) for
Dutch. An example can be found below.

(2) Van Gogh is een beroemde schilder.
Van Gogh is a famous painter.

We extracted around 342K categorised named entity types overall distributed
over some 180K named entities. 90.6% of the data is found using the apposition
relation and 9.4% is found scanning the corpus for predicate complements.

This database contains, for instance, 391 names of islands (Bali, Bonaire,
Aruba etc.) and 186 different queens (Elizabeth, Wilhelmina, Beatrix etc.). The
class labels extracted for each named entity may contain a certain amount of
noise. However, by focusing on the most frequent label for a named entity, most
of the noise can be discarded. For instance, Beatrix occurs 1,210 times in the
extracted tuples, 1,150 times as queen, and not more than 60 times with various
other labels (play, name, hat, possibility, ...).

Regarding the ambiguity of the classified named entities we can say that on
average a named entity has 1.9 labels. The distribution is skewed: 80% has only
one label and for example the most ambiguous named entity, the Netherlands,
has 704 labels in total.

In previous work we used the categorised named entities as they were re-
trieved from the CLEF corpus (80M words). The experiments reported in sec-
tion 6.7 and parts of section 6.6, which are both based on this work, use this
list of categorised named entities.

When larger corpora became available, we made use of this data. We used
the data of the TwNC corpus (500M words) and Dutch Wikipedia (50M words)
to extract apposition relations. This made the problem of the skewed data even
worse. The Netherlands now appears with 1,251 different labels. To filter out
incorrect and highly unlikely labels (often the result of parsing errors) we tried
several association measures, e.g. Pointwise Mutual Information (Church and
Hanks, 1989) and \(t\)-test. However, these association measures did not give us
what we wanted.

For example, the actress Audrey Hepburn, is found with the label actrice ‘ac-
tress’ 18 times, 6 occurrences with filmster ‘movie star’ and 4 occurrences with

\(^3\)We limited our search to the predicate complement relation between named entities and
a noun and excluded examples with negation.
6.3. Lexico-semantic information used

chauffeursdochter ‘daughter of a chauffeur’. The named entity is only found 3 times with the label Unicef-ambassadrice ‘ambassador for Unicef’, verkoopster ‘sales woman’, and ster ‘star’. The t-test and PMI both select Unicef-ambassadrice ‘ambassador for Unicef’ as the most important association. Although actress and movie star are very important labels for the named entity Audrey Hepburn, they appear on the 10th and 8th position when using Pointwise Mutual Information as an association measure.

This is to be expected, since the label actrice ‘actress’ appears many times in the corpus with various other named entities, whereas Unicef-ambassadrice ‘ambassador for Unicef’ is much less frequent. This will be taken into account by association measures such as Pointwise Mutual Information. However, the fact that the label actrice ‘actress’ appears frequently in general does not mean that it is not a good label for the named entity Audrey Hepburn.

We therefore chose to look at the relative frequency of the combination of the named entity and a category with regard to the frequency of the named entity overall. We divided the frequency of the co-occurrence of the named entity and the category by the frequency of the named entity in general. We set a threshold of 0.05. All categorised named entities with relative frequencies under 0.05 were discarded. The most frequent named entity Nederland ‘The Netherlands’, which previously had 1251 labels in total has none after applying this cutoff. This is a drawback of using the filtering with this cut-off. However, the number of unwanted labels is considerably lower.

For Audrey Hepburn the last 5 categories of the following list are removed:

<table>
<thead>
<tr>
<th>Audrey Hepburn:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>actrice</td>
<td>‘actress’</td>
</tr>
<tr>
<td>filmster</td>
<td>‘movie star’</td>
</tr>
<tr>
<td>chauffeursdochter</td>
<td>‘daughter of a driver’</td>
</tr>
<tr>
<td>ster</td>
<td>‘star’</td>
</tr>
<tr>
<td>Unicef-ambassadrice</td>
<td>‘ambassador for Unicef’</td>
</tr>
<tr>
<td>verkoopster</td>
<td>‘sales woman’</td>
</tr>
<tr>
<td>secretaresse</td>
<td>‘secretary’</td>
</tr>
<tr>
<td>man</td>
<td>‘man’</td>
</tr>
<tr>
<td>filmlegende</td>
<td>‘film legend’</td>
</tr>
<tr>
<td>cockney-bloemenmeisje</td>
<td>‘cockney flower girl’</td>
</tr>
</tbody>
</table>

We would have rather saved filmlegende ‘film legend and thrown away chauffeursdochter ‘daughter of a driver’, but the filtering based on relative frequencies allows us only to set a threshold and not to change the ranking of the labels.

For Monica Seles we are left with the first 4 categories of the following list:
Chapter 6. Using lexico-semantic knowledge for question answering

Monica Seles:

<table>
<thead>
<tr>
<th>Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>tennissster</td>
<td>‘tennis star’</td>
</tr>
<tr>
<td>landgenote</td>
<td>‘compatriot’</td>
</tr>
<tr>
<td>winnares</td>
<td>‘winner’</td>
</tr>
<tr>
<td>tennisspeelster</td>
<td>‘tennis player’</td>
</tr>
<tr>
<td>wereld</td>
<td>‘world’</td>
</tr>
<tr>
<td>speelster</td>
<td>‘player’</td>
</tr>
<tr>
<td>rivale</td>
<td>‘rival’</td>
</tr>
<tr>
<td>nummer</td>
<td>‘number’</td>
</tr>
<tr>
<td>koppel</td>
<td>‘duo’</td>
</tr>
<tr>
<td>finale</td>
<td>‘finals’</td>
</tr>
<tr>
<td>antwoord</td>
<td>‘answer’</td>
</tr>
</tbody>
</table>

6-7
6-4
6-4

Also here we would have liked to keep speelster ‘player’ and rivale ‘rival’, but we are happy to get rid of most of the labels from 5 to 14.

After applying this cutoff, we are left with 382K categorised named entity types for 218K named entities. Regarding the ambiguity of the classified named entities, we can say that on average a named entity has 1.75 labels. The distribution is less skewed: 67% has one label and for example the most ambiguous named entity, Twinning, has 18 labels in total. In Table 6.1 we see the amount of information that is contained in the list of categorised named entities of previous work, Cat. NEs(1), and the new filtered list of categorised named entities, Cat. NEs(2).

We use this larger set of filtered categorised named entities in section 6.5 and parts of section 6.6.

6.4 Question analysis (case study)

6.4.1 Introduction

The task of question analysis is to determine the question type and to identify keywords in the question. Questions are classified according to the expected answer type. There are several question types, such as location for questions asking for a location and measure for questions asking for the size or length of something. The question type that we will be concerned with here is the question type function and a variation of the function question: the function_of question. In (3) examples are given for the two question types.
6.4. Question analysis (case study)

(3) Wie is de Noorse bondscoach? ← function
    ‘Who is the Norwegian national team coach?’

Van welke Franse voetbalclub was Bernard Tapie voorzitter? ← funct_of
    ‘Of which French football club was Bernard Tapie president?’

These questions are in most cases answered by means of table-lookup. This means that we extract functions of people and their names from the document collection beforehand to be able to answer questions like these easily during the question answering process. To be able to classify the question *Who is the Norwegian national team coach?* as a function question we must know that *national team coach* is a function people have. This is where lexico-semantic information is needed. We need a list of functions people have to correctly classify the question as a function question.

To obtain a list of words describing a role or function, we extracted from Dutch EWN all words under the node *leider* (leader). These are 255 nouns in total. The majority of hyponyms of this node seemed to indicate function words we were interested in, i.e. it contained the Dutch equivalents of *king, queen, president, director, chair*, etc., while other potential candidate nodes, such as *beroep* ‘profession’ seemed less suitable. However, the coverage of this list, when tested on a newspaper corpus, is far from complete. On the one hand, the list contains a fair number of archaic items, while on the other hand, many functions that occur frequently in newspaper text are missing, i.e. Dutch equivalents of *banker, boss, national team coach, captain, secretary-general* etc.

To improve recall we decided to expand the list. We want to expand the list with co-hyponyms in order to find other types of functions people have. As the syntax-based method is particularly good at finding co-hyponyms, we used the syntax-based nearest neighbours to expand this list. In section 6.4.4 we further explain how we did this.

6.4.2 Description of component

During question analysis the question type is identified and important keywords in the question are selected. In order to determine the question type, dependency patterns are written to be matched against the dependency analysis of the question.

The dependency analysis of a sentence gives rise to a set of dependency

---

4The hyponym structure of EWN is complex. There are more labels that categorise more or less for the function relation. We chose the label *leader*, because it seemed to be the least noisy.
relations of the form \( \langle \text{Head/HIx, Rel, Dep/DIx} \rangle \), where \text{Head} is the root form of the head of the relation, and \text{Dep} is the head of the dependent. \text{HIx} and \text{DIx} are string indices, and \text{Rel} the dependency relation. For instance, the dependency analysis of sentence (4-a) is (4-b).

\begin{align*}
(4) & \quad \text{a. Wie is de bondscoach van Noorwegen?} \\
& \quad \text{‘Who is the national team coach of Norway?’} \\
& \quad \{ \langle \text{ben/2, su, wie/1} \rangle, \\
& \qquad \langle \text{bondscoach/4, predc, ben/2} \rangle, \\
& \qquad \langle \text{bondscoach/4, mod, van/5} \rangle, \\
& \qquad \langle \text{van/5, obj1, Noorwegen/6} \rangle \}\end{align*}

A dependency pattern is a set of (partially underspecified) dependency relations:

\begin{align*}
(5) & \quad \{ \langle \text{ben/A, su, wie/B} \rangle, \\
& \qquad \langle \text{Function/C, predc, ben/A} \rangle, \\
& \qquad \langle \text{Function/C, mod, van/D} \rangle, \\
& \qquad \langle \text{van/D, obj1, Named_Entity/E} \rangle \}\end{align*}

A pattern may contain variables, represented here by words starting with a capital, such as \text{Function}.

The aim of this study is to determine the list of words that can fill the slot of the \text{Function} variable.

### 6.4.3 Related work

Lexico-semantic resources have been used by various QA teams to classify questions in QA systems. Pasca and Harabagiu (2001) have used a semi-automatically built answer type taxonomy for answer type recognition. It encodes 8,707 WordNet synsets, 20 tops (these are the top-most answer type categories) and 129 manually added links. They report 75% of the 893 TREC evaluation questions to be correctly recognised, when using this answer taxonomy. Building an answer type taxonomy semi-automatically from WordNet proved beneficial for question classification.

Automatically acquired clusters of semantically related words can be used to extend or enrich existing ontological resources. Alfonseca and Manandhar (2002), for instance, describe a method for expanding WordNet automatically. New concepts are placed in the WordNet hierarchy according to their distributional similarity to words that are already in the hierarchy. Their algorithm performs a top-down search and stops at the synset that is most similar to the new concept.
6.4.4 Methodology

To improve recall, we extended the list of function words obtained from EWN semi-automatically with distributionally similar words from the syntax-based method. In particular, for each of the 255 words in the EWN list, we retrieved its 100 nearest neighbours. We gave each retrieved word a score that corresponds to its reverse rank (1st word: 100, 2nd: 99, 3rd: 98 etc.). The overall score for a word was the sum of the scores it obtained for the individual target words. Thus, words that are semantically similar to several words in the original list obtain a higher score than words that were returned only once or twice. Words that were present already in the EWN-list were filtered.

An informal evaluation of the result showed that many false positives in the expanded list were either named entities or nouns referring to groups of people, e.g. board, committee. The distinction between groups and functions of individuals is hard to make on the basis of syntax-based distributional data. For instance, both a board and a director can take decisions, report results, be criticised etc. We tried to filter both proper names and groups automatically, by discarding noun stems that start with a capital, and noun stems which are listed under the node group (group) in EWN. Finally, we selected the top-1000 of the filtered list, and validated it manually. The list contained 644 valid role or function nouns, which are absent in EWN. A substantial number of the errors are still nouns that refer to a group, but that are not listed as such in EWN.

The 644 valid nouns were merged with the original EWN list, to form a list of 899 function or role nouns.

6.4.5 Evaluation

We evaluated the performance of question analysis on function and function of questions, using the original EWN list and the semi-automatically expanded list with the help of nearest neighbours from the syntax-based method. We evaluated on the CLEF ('03, '04, '05) Dutch QA test set (approximately 775 questions). CLEF is the Cross-Language Evaluation Forum, a framework for testing, tuning, and evaluation of information retrieval systems operating on European languages, hence its name. It provides a test bed for question answering systems in multiple European languages.

We calculated for each run the Mean Reciprocal Rank (MRR) and the CLEF score. The MRR measures the percentage of passages for which a correct answer was found in the top-k passages returned by the system. The MRR score is the average of 1/R where R is the rank of the first relevant passage computed over the 5 highest ranked passages only. Passages retrieved were considered relevant.

\(^{5}\text{http://clef-qa.itc.it}\)
Chapter 6. Using lexico-semantic knowledge for question answering

<table>
<thead>
<tr>
<th>Q-type</th>
<th>Baseline</th>
<th>EWN</th>
<th>EWN+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># q</td>
<td>MRR</td>
<td>CLEF</td>
</tr>
<tr>
<td>Funct</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Funct.of</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WH</td>
<td>114</td>
<td>0.49</td>
<td>0.45</td>
</tr>
<tr>
<td>Person</td>
<td>117</td>
<td>0.75</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Total</td>
<td>775</td>
<td>0.66</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 6.2: Overall performance of the baseline system and the EWN and the expanded EWN+ system on the CLEF ('03, '04, '05) Dutch QA test set.

when one of the possible answer strings is found in that passage. The CLEF score gives the precision of the first (highest ranked) answer only.

6.4.6 Results

Expanding the list of functions from EWN semi-automatically results in 17 questions being classified as function or function.of questions in addition to the 71 questions selected on the basis of the EWN function list only. Functions that were missing in the EWN list were general functions, such as adviseur ‘advisor’, bestuursvoorzitter ‘board secretary’, bondscoach ‘national team coach’, opvolger ‘successor’, but also functions in family such as broer ‘brother’, vader ‘father’, weduwe ‘widow’ etc.

With respect to introducing Function and function.of question types we can see in Table 6.2 that the introduction of these question types results in large improvements. The classification has become more fine-grained. Therefore the questions can be answered more adequately. Person questions are already answered relatively well, while on the other hand WH questions are answered with a much lower precision. Shift from person and WH questions to function and function.of questions is beneficial. However, the differences between using the EWN function list and the expanded function list that has 644 valid function nouns in addition are minimal. There are 9 questions that receive a different answer of which 4 receive a lower score and 5 receive a higher score.

6.4.7 Conclusion

Question analysis is improved by expanding the list of functions from EWN semi-automatically with nearest neighbours from the syntax-based method. A total of 88 questions are classified as function or function.of questions, whereas only 71 questions were selected on the basis of the EWN function list.

We showed that adding a question class for functions and function.of
results in a shift from person and WH questions to function and function_of
questions that is beneficial for the overall performance of the system. Using
EWN or the expanded EWN list does not result in large differences for the QA
system as a whole.

6.5 Query expansion for passage retrieval

6.5.1 Introduction

Information retrieval (IR) is used in most QA systems to filter out relevant
passages from large document collections to narrow down the search for answer
extraction modules in a QA system. Accurate IR is crucial for the success of
this approach. Answers in paragraphs that have been missed by IR are lost for
the entire QA system. Hence, high performance of IR especially in terms of
recall is essential. Furthermore, high precision is desirable as IR scores are used
for answer extraction heuristics and also to reduce the chance of subsequent
extraction errors.

Because the user’s formulation of the question is only one of the many pos-
sible ways to state the information need that the user might have, there is often
a discrepancy between the terminology used by the user and the terminology
used by the author to describe the same concept. A document might hold the
answer to the user’s question, but it will not be found due to the TERMINO-
LOGICAL GAP. Moldovan et al. (2002) show that their system fails to answe-
ring many questions (25.7%) because of the terminological gap, i.e. keyword expan-
sion would be desirable but is missing. Query expansion techniques have been
developed to bridge this gap. We will give an overview of the various methods
that have been used in the section on related work (6.5.3).

We hope that the synonyms retrieved automatically, and in particular the
synonyms retrieved by the alignment-based method, as these are most precise,
will help to overcome this terminological gap.

However, we believe that there is more than just a terminological gap. There
is also a KNOWLEDGE GAP. Documents are missed or do not end up high in the
ranks because additional world knowledge is missing. We are not speaking of
synonyms here, but words belonging to the same subject field. When a user is
looking for information about the explosion of the first atomic bomb, mentally
a subject field is active that could include: war, disaster, and World War II.

Regarding the several types of lexico-semantic information we have, we ex-
pect that the proximity-based method would be most helpful to overcome the
knowledge gap. We expect that, if its loosely structured data and associative
nature would be helpful in any component, it would be the IR component. Of-
ten the associations given for a word already include the answer. For example, for the proper name Melkert, a former Dutch minister of social affairs and employment, we get the association werkgelegenheid ‘employment’. This type of information can be very helpful in IR query expansion for a question such as Who is Ad Melkert?.

The other components, such as question analysis and answer matching and selection, need much more structured information. The fact that the proximity-based method retrieves nearest neighbours that are of a different syntactic category than the headword is not harmful for the IR component, since it uses a bag-of-words method. It is, however, less suitable, for instance, for answer matching, where we are matching dependency relations. We want to include variations such as infant for child in a question such as How much should an infant eat? We do not want to include associations, such as diaper or crying, when matching the answer and the question’s dependency relations regarding infant.

Apart from the proximity-based method, extra information, such as the semantic category of named entities, can be very helpful to overcome the knowledge gap. Knowing that Monica Seles is a tennis player helps to find relevant passages regarding this tennis star.

6.5.2 Description of component

In Bouma et al. (2007) linguistic information is exploited as a knowledge source for IR. Several layers of linguistic features and feature combinations extracted from syntactically analysed sentences are defined and included as index fields in the IR component. Although the linguistic information improves the scores considerably, we have chosen not to use this type of information in the present experiments. The addition of multiple layers of information makes it much harder to see what the contribution of the expansions is.

In the experiments presented in this section we take a simple bag-of-words approach, where root forms are used instead of words. We used the IR-system Lucene from the Apache Jakarta project (Jakarta, 2004). Lucene is a widely-used open-source Java library with several extensions and useful features. We apply standard settings and stop word removal.

\footnote{This to make the matching with the expansions resulting from the three distributional methods that are all root forms easier. Lastly, we used the Alpino dependency parser to retrieve root forms.}
6.5.3 Related work

There are many ways to expand queries and expansions can be acquired from several sources. For example, one can make use of collection-independent knowledge structures, such as WordNet. In contrast, collection-dependent knowledge structures are often constructed automatically based on data from the collection. For example, by extracting lexico-semantic information using distributional methods on the texts collection. Expansion methods based on such collection-dependent knowledge structures are also referred to as global techniques. Relevance feedback is an approach that modifies the initial query using words from documents retrieved by the system. The user selects the most relevant documents from a top-ranked list and terms from these documents are added to the original query. When applying pseudo relevance feedback (also known as automatic, blind, or ad-hoc relevance feedback) there is no user intervention. The top-ranked documents are used directly to expand the original query. Expansion techniques that use the top-ranked documents for expansion are called local techniques.

We will discuss some examples of these approaches. If available, we will discuss methods applied in the context of QA systems.

Monz (2003) ran experiments using blind or pseudo relevance feedback for IR in a QA system. The author reports dramatic decreases in performance. He argues that this might be due to the fact that there are usually only a small number of relevant documents. Another reason he gives is the fact that he used the full document to fetch expansion terms and the information that allows one to answer the question is expressed very locally.

A global technique that is most similar to ours uses syntactic context to find suitable terms for query expansion (Grefenstette, 1992, 1994a). The author reports that the gain is modest: 2% when expanded with nearest neighbours found by his system and 5 to 6%, when applying stemming and a second loop of expansions of words that are in the family of the augmented query terms. Although the gain is greater than when using document co-occurrence as context, the results are mixed, with expansions improving some query results and degrading others.

Another source for finding expansions are existing hand-built corpus-independent thesauri. Moldovan et al. (2003) show in a detailed error analysis that 25.7% of the errors are due to the fact that keyword expansion would be desirable, but is missing, i.e. due to the terminological gap. By using a lexico-semantic feedback loop that feeds lexico-semantic alternations from WordNet as keyword expansions to the retrieval component, the MRR score of their system is improved by

\footnote{i.e. words that appear in the same documents and that share the first three, four or five letters.}
Chapter 6. Using lexico-semantic knowledge for question answering

Pasca and Harabagiu (2001) use lexico-semantic information from WordNet in two different ways. First, they use the information for keyword alternation on the morphological, lexical (synonyms and other related words) and semantic level (no synonyms, but there exists a chain of relations in WordNet between the two words). They evaluated their system on question sets of TREC-8 and TREC-9. For TREC-8 they reach a precision score of 55.3% without including any alternations for question keywords, 67.6% if lexical alternations are allowed and 73.7% if both lexical and semantic alternations are allowed. Morphological alternations increase the precision scores by 3.5% on a separate test set (115 questions) from TREC-9. Second, they extract from WordNet knowledge about the specificity of the question keywords by counting its hyponyms. If the count is smaller than a certain threshold (here 10), the word is deemed very specific and should not be dropped from the search process. The other way around, words that are not specific enough should be dropped, such as city in What city is the capital of the United Kingdom?. The heuristics make the number of correctly answered questions increase from 133 (65%) to 151 (76%) on the test set of TREC-8.

However, Yang and Chua (2003) report that adding additional terms from WordNet’s synsets and glosses adds more noise than information to the query. Also, Voorhees (1993) concludes that expanding by automatically generated synonym sets from EWN can degrade results. In Yang et al. (2003) the authors use external knowledge extracted from WordNet and the Web to expand queries for QA. Minor improvements are attained when the Web is used to retrieve a list of nearby (one sentence or snippet) non-trivial terms. When WordNet is used to rank the retrieved terms the improvement is reduced. The best results are reached when structure analysis is added to knowledge from the Web and WordNet. Structure analysis determines the relations that hold between the candidate expansion terms to identify semantic groups. Semantic groups are then connected by conjunction in the Boolean query.

The approach by Qiu and Frei (1993) is a global technique. They automatically construct a similarity thesaurus, based on the documents terms appear in. They use word-by-document matrices, where the features are document IDs, to determine the similarity between words. Expansions are selected based on the similarity to the query concept, i.e. all words in the query together, and not based on the single words in the query independently. The results they get are promising. As far as we know, the method has not been tested for IR in a QA setting.

Pantel and Ravichandran (2004) have used a method that is not related to query expansion but yet very related to our work. They have semantically
indexed the TREC-2002 IR collection with the isa-relations found by their system for 179 questions that had an explicit semantic answer type, such as *What band was Jerry Garcia with?*. They show small gains in the performance of the IR output using the semantically indexed collection.

The experiments in this chapter are partly based on global techniques. The proximity-based method uses the same corpus as is used for document retrieval, the CLEF corpus. The syntax-based method uses the TwNC corpus, of which the CLEF corpus is a subset. We have, however, also used corpus-independent knowledge sources. The expansions resulting from the alignment-based method are retrieved from the Europarl corpus. In addition, we use the synsets of Dutch EWN.

### 6.5.4 Methodology

In order to test the performance of the three distributional methods on query expansion for passage retrieval, we ran several tests. The baseline is running Lucene on root forms with standard settings and stop word removal.

We applied the nearest neighbours resulting from the three methods as described in section 6.3:

- Nearest neighbours of syntax-based distributional similarity
- Nearest neighbours of alignment-based distributional similarity
- Nearest neighbours of proximity-based distributional similarity

For all methods we selected the top-5 nearest neighbours that had a similarity score of more than 0.2 as expansions.

Apart from the three distributional methods we ran experiments using:

- EuroWordNet
- Categorised named entities (Cat. NEs(2), as described in section 6.3)

For EWN all words in the same synset (for all senses) were added as expansions. We do not have similarity scores for EWN and thus did not use a threshold here. The categorised named entities were not only used to expand named entities with the corresponding label, but also to expand nouns with named entities. In the first case all labels were selected. As we have seen in section 6.3, the maximum is not more than 18 labels anyway. In the second case some nouns get many expansions. For example, a noun, such as *vrouw* ‘woman’, gets 1,751 named entities as expansions. We discarded nouns with more than 50 expansions, as these were deemed too general and hence not very useful.
Chapter 6. Using lexico-semantic knowledge for question answering

<table>
<thead>
<tr>
<th></th>
<th>SynCat</th>
<th>EWN</th>
<th>Syntax</th>
<th>Align</th>
<th>Proxi</th>
<th>Cat.NEs</th>
<th>Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>51.52</td>
<td>51.15</td>
<td>51.21</td>
<td>51.38</td>
<td>51.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adj</td>
<td>52.33</td>
<td>52.27</td>
<td>52.38</td>
<td>51.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verbs</td>
<td>52.40</td>
<td>52.33</td>
<td>52.21</td>
<td>52.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proper</td>
<td>52.59</td>
<td>50.16</td>
<td></td>
<td>53.94</td>
<td>55.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>51.65</td>
<td>51.21</td>
<td>51.02</td>
<td>53.36</td>
<td>55.29</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: MRR scores for the IR component with query expansion from several sources

The last two settings are the same for the expansions resulting from distributional methods and the last two types of lexico-semantic information.

- Expansions were added as root forms
- Expansions were given a weight such that all expansions for one original keyword add up to 0.5.

6.5.5 Evaluation

For evaluation we used data collected at the CLEF competitions on Dutch QA. The CLEF text collection contains 4 years of newspaper text, approximately 80 million words and Dutch Wikipedia, approximately 50 million words. We used the question sets from the competitions of the Dutch QA track in 2003, 2004, and 2005. Questions in these sets are annotated with valid answers found by the participating teams including IDs of supporting documents in the given text collection. We expanded these list of valid answers where necessary.

We calculated for each run the Mean Reciprocal Rank (MRR). The MRR measures the percentage of passages for which a correct answer was found in the top-$k$ passages returned by the system. The MRR score is the average of $1/R$ where $R$ is the rank of the first relevant passage computed over the 20 highest ranked passages. This is in contrast to the MRR scores given in section 6.4.5, where only the 5 highest ranked passages were taken into account. Passages retrieved were considered relevant when one of the possible answer strings was found in that passage.

6.5.6 Results

In Table 6.3 the MRR (Mean Reciprocal Rank) is given for the various expansion techniques. We have given MRR scores for expanding the several syntactic categories, where possible. We were not able to include proper names for the
6.5. Query expansion for passage retrieval

<table>
<thead>
<tr>
<th>SynCat</th>
<th>EWN (+/-)</th>
<th>Syntax (+/-)</th>
<th>Align (+/-)</th>
<th>Proxi (+/-)</th>
<th>Cat.NEs (+/-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>27/50</td>
<td>28/61</td>
<td>17/58</td>
<td>64/87</td>
<td>17/37</td>
</tr>
<tr>
<td>Adj</td>
<td>3/6</td>
<td>1/2</td>
<td>1/2</td>
<td>31/47</td>
<td></td>
</tr>
<tr>
<td>Verbs</td>
<td>31/51</td>
<td>5/10</td>
<td>8/32</td>
<td>51/56</td>
<td></td>
</tr>
<tr>
<td>Proper</td>
<td>3/2</td>
<td>30/80</td>
<td></td>
<td>76/48</td>
<td>157/106</td>
</tr>
<tr>
<td>Overall</td>
<td>56/94</td>
<td>56/131</td>
<td>25/89</td>
<td>161/147</td>
<td>168/130</td>
</tr>
</tbody>
</table>

Table 6.4: Number of questions that receive a higher (+) or lower (-) RR when using expansions from several sources

alignment-based method, due to decisions we made earlier regarding pre-processing.\footnote{8}{All words were transformed to lowercase during pre-processing. The inclusion of these lowercase words gives rise to too much ambiguity. If applied correctly, we believe that the alignment-based method could gather spelling variations of names.} The baseline does not make use of any expansion for any syntactic category.

In Table 6.4 the number of questions that get a higher versus the number of questions that get a lower reciprocal rank (RR) after applying the individual lexico-semantic resources are given. For example, the adjectival expansions from the syntax-based method result in 1 question that gets a higher RR score (compared to the baseline) and 2 questions that get a lower RR score. Apart from expansions on adjectives and proper names from EWN the impact of the expansions is substantial.

The fact that adjectives have so little impact is due to the fact that there are not many adjectives among the query terms.\footnote{9}{Moreover the adjectives related to countries, such as German and French and their expansion Germany, France are handled by a separate list.}

The negligible impact of the proper names from EWN is surprising since EWN provides more entries for proper names than the proximity-based method (1.4K vs 1.2K, as can be seen in 6.1). The proximity-based method clearly provides information about proper names that are more relevant for the corpus used for QA, as it is built from that same corpus. This shows the advantage of using corpus-based methods. The impact of the expansions resulting from the syntax-based method lies in between the two previously mentioned expansions. It uses a corpus of which the corpus used for QA is a subset. The type of expansions that result from the proximity-based method have a larger effect on the performance of the system than those resulting from the syntax-based method.

For most of the resources the number of questions that show a rise in RR is smaller than the number of questions that receive a lower RR, except for the expansion of proper names by the categorised named entities and the proximity-
Chapter 6. Using lexico-semantic knowledge for question answering

The categorised named entities provide the most successful lexico-semantic information, when used to expand named entities with their category label. It is clear that using the same information (the categorised named entities) in the other direction, i.e. to expand nouns with named entities of the corresponding category hurts the scores. We know from Table 6.1 in section 6.3 that this resource has 70 times more data than the proximity-based resource. Also, we would like to remind the reader of a problem with the task-based evaluation that we discussed in the last subsection of section 2.5.2: The test sets of the CLEF testbeds are not as much motivated by what users might want to ask, but rather by what question answering systems are currently able to handle, e.g. factoid questions. The fact that the categorised named entities and expansions of proper names give rather positive results can be partly attributed to the nature of the questions in the CLEF test sets we use for the evaluation.

The proximity-based expansions are rather important as well. The most important reason for this might be the fact that we used the 50K most frequent words as headwords only. The nearest neighbours resulting from the proximity-based method are therefore always among the 50K most frequent words. The probability that these words are found in documents is higher than for perhaps less frequent expansions resulting from the other methods, where we did not set a threshold except the exclusion of hapaxes.

The expansions resulting from the syntax-based method do not result in any improvements. As expected, the expansion of proper names from the syntax-based method hurts the performance most. The MRR drops from 52.36% to 50.16%. Remember that the nearest neighbours of the syntax-based method often include co-hyponyms. For example, Germany would get The Netherlands and France as nearest neighbours. It does not seem to be a good idea to expand the word Germany with The Netherlands and France, when a user, for example, asks for the name of the Minister of Foreign Affairs of Germany.  

Remember from the introduction of this section (6.5.1) that we made a distinction with regard to missing information in queries. We referred to the phenomenon that documents are missed due to differences in wording between the user’s query and the document containing the answer by the term ‘terminological gap’. Expansions in the form of extra information about the subject field of the query that result in more relevant documents being retrieved were referred to by the term ‘knowledge gap’. The lexico-semantic resources that are suited to bridge the terminological gap, such as synonyms from the alignment-based method and EWN, do not result in improvements in the experiments under discussion. For all syntactic categories either the same scores or slightly lower scores are attained. However, the lexico-semantic resources that may be used to bridge the knowledge gap, i.e. associations from the proximity-based
method and categorised named entities, do result in improvements of the IR component.

Let us first take a look at the disappointing results regarding the terminological gap, before we move to the more promising results related to the knowledge gap. We expected that the expansions of verbs would be particularly helpful to overcome the terminological gap, which is large for verbs, since there is much variation. We will give some examples of expansion from the alignment-based method and EWN.

(6) Wanneer werd het Verdrag van Rome getekend?
‘When was the treaty of Rome signed?’

<table>
<thead>
<tr>
<th>Expansions for teken ‘sign’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align</td>
</tr>
<tr>
<td>typeer ‘typify’</td>
</tr>
<tr>
<td>onderteken ‘sign’</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

For the example in (6) both the alignment-based expansions and the expansion from EWN result in a decrease in RR of 0.5. The verb teken ‘sign’ is ambiguous. We see three senses of the verb represented in the EWN list, i.e. drawing, characterising, and signing as in signing an official document. One out of the two expansions for the alignment-based method and 2 out of 9 for EWN are in principle synonyms of teken ‘sign’ in the right sense for this question. However, the documents that hold the answer to this question do not use synonyms for the word teken. The expansions only introduce noise.

We found a positive example in (7). The RR score is improved by 0.3 for both the alignment-based expansions and the expansions from EWN, when expanding explodeer ‘explode’ with ontplof ‘blow up’.

(7) Waar explodeerde de eerste atoombom?
‘Where did the first nuclear bomb explode?’
Expansions for *explode* ‘explode’

<table>
<thead>
<tr>
<th>Align</th>
<th>EWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontplof ‘blow up’</td>
<td>barst los ‘burst’</td>
</tr>
<tr>
<td>ontplof ‘blow up’</td>
<td>ontplof ‘blow up’</td>
</tr>
<tr>
<td>barst uit ‘crack’</td>
<td></td>
</tr>
<tr>
<td>plof ‘boom’</td>
<td></td>
</tr>
</tbody>
</table>

To get an idea of the amount of terminological variation between the questions and the documents, we determined the optimal expansion words for each query, by looking at the words that appear in the relevant documents. When inspecting these, we learned that there is in fact little to be gained by terminological variation. In the 25 questions we inspected we found 1 near-synonym only that improved the scores: *gekke-koeienziekte* ‘mad cow disease’ → *Creutzfeldt-Jacobziekte* ‘Creutzfeldt-Jacob disease’.

The fact that we find only few synonyms might be related to a point noted in Chapter 2, when we discussed the problems related to evaluating on a task such as open-domain QA (section (4)). Mur (2006) showed proof that some of the questions in the CLEF track that we use for evaluation, have the looks of back formulations. Although Magnini et al. (2004) claim that the questions are made independently of the document collection, the example Mur (2006) gives is rather convincing. This means that the poor results presented in this section might be misleading. A more natural setting might have given rise to more terminological variation and hence better evaluation results for the methods that try to overcome the terminological gap.

After inspecting the optimal expansions, we were under the impression that most of the expansions that improved the scores were related to the knowledge gap, rather than the terminological gap. Now the expansions related to the knowledge gap are very broad in nature, as is the case in general with associations. A word’s associations are unlimited, whereas its synonyms are finite. Moreover, some words do not have synonyms at all. The difficulty with bridging the knowledge gap is selecting the relevant background knowledge for a query. We will now give some examples of good and bad expansions related to the knowledge gap.

The categorised named entities result in the best expansions, followed by the proximity-based expansions. In (8) an example is given for which categorised named entities proved very useful:

(8) Wie is Keith Richard?
   ‘Who is Keith Richard?’
Expansions for *Keith Richard*

<table>
<thead>
<tr>
<th>Cat. NEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>gitarist ‘guitar player’</td>
</tr>
<tr>
<td>lid ‘member’</td>
</tr>
<tr>
<td>collega ‘colleague’</td>
</tr>
<tr>
<td>Rolling Stones-gitarist ‘Rolling Stones guitar player’</td>
</tr>
<tr>
<td>Stones-gitarist ‘Stones guitar player’</td>
</tr>
</tbody>
</table>

It is clear that this type of information helps a lot in answering the question in (8). It contains the answer to the question. The RR for this question goes from 0 to 1. We see the same effect for the question *Wat is NASA?* ‘What is NASA?’.

It is a known fact that named entities are an important category for QA. Many questions ask for named entities or facts related to named entities. From these results we can see that adding the labels that are related to named entities are useful for retrieving the passages that contain the answer.

Note, that the proper names from the categorised named entities include multi-word expressions, such as *Keith Richard*, because they result from the syntax-based method. This is different from the proper names resulting from the proximity-based method, where no multi-word terms are included. The proximity-based proper name expansions are limited to the 1,399 most frequent proper names, as explained in section 6.3. The categorised named entities result from the automatically filtered appositions from Wikipedia and the TwNC corpus. The list comprises category labels for approximately 218K named entities. We can see that the large list of the categorised named entities results in more questions being expanded in Table 6.4.

Apart from differences in size and handling of multi-word terms, the expansions retrieved are different. The expansions resulting from categorised named entities are all category labels. We will show some examples of expansions from these two sources.

For example, Rwanda gets different expansions from the two methods. Consider the question in (9).

(9) Welke bevolkingsgroepen voerden oorlog in Rwanda?
    ‘What populations waged war in Rwanda?’
In this case the expansions from the proximity-based method are very useful (except for Zaire), since they include the answer to the question. That is not always the case, as can be seen in (10). However, the expansions from the categorised named entities are not very helpful in this case either.

(10) Wanneer werd het Verdrag van Rome getekend?
‘When was the treaty of Rome signed?’

IR does identify Verdrag van Rome ‘Treaty of Rome’ as a multi-word term, however, it adds the individual parts of multi-word terms as keywords as a form of compound analysis. It might be better to expand the multi-word term only and not its individual parts to decrease ambiguity. Verdrag van Rome ‘Treaty of Rome’ is not found in the proximity-based nearest neighbours because it does not include multi-word terms.

Still, it is not very helpful to expand the word Rome with pope for this question that has nothing to do with religious affairs. We can see this as a problem of word sense disambiguation. The association pope belongs to Rome in the religious sense, the place where the Catholic Church is seated. Rome is often referred to as the Catholic Church itself, as in Henry VIII broke from Rome. Gonzalo et al. (1998) showed in an experiment, where words were manually disambiguated, that a substantial increase in performance is obtained when query words are disambiguated, before they are expanded.

We tried to take care of these ambiguities by using an overlap method. The overlap method selects expansions that are found in the nearest neighbours of
more than two query words. The method is related, though much simpler than the method used by Qiu and Frei (1993) discussed in section 6.5.3. Unfortunately, as Navigli and Velardi (2003) note, who implement a similar technique, using lexico-semantic information from WordNet, the common nodes expansion technique works very badly. Also, Voorhees (1993) who uses a similar method to select expansions concludes that the method has the tendency to select very general terms that have more than one sense themselves. In future work we would like to implement the method by Qiu and Frei (1993) for the proximity-based method, which uses a more sophisticated technique to combine the expansions of several words in the query.

6.5.7 Conclusion

We can conclude from these experiments on query expansion for passage retrieval that query expansion with synonyms to overcome the terminological gap is not very fruitful. We believe that the noise introduced by ambiguity of the query terms is more important than the positive effect of adding lexical variants. This is in line with findings by Yang and Chua (2003). On the contrary, Paşca and Harabagiu (2001) were able to improve their QA system by using lexical and semantic alternations from WordNet. This might be due to the architecture of their system in which feedback loops are used. This means that question keywords are expanded only when there is a reason to activate the feedback loop. For example, when there are no matching terms between question and candidate answer context. In general it is hard to compare system improvements because the baselines used are not equal. Our baseline system has quite a number of tricks that try to overcome problems caused by terminological variation, such as, heuristics taking the answer context into account.

However, adding extra information with regard to the subject field of the query, query expansions that bridge the knowledge gap, proved slightly beneficial. The proximity-based expansions augment the MRR scores with 1.5%. Most successful are the categorised named entities. These expansions were able to augment the MRR scores with nearly 3.5%. Grefenstette (1994a) reports higher scores for the syntax-based method than for the unstructured document co-occurrence-based method. Our unstructured approach is based on sentences instead of documents and is thus more fine grained, which might explain the improvements. Monz (2003) noted that using documents to fetch expansion terms might be less suitable as the information that allows one to answer the question is often expressed very locally.
6.6 Answer matching and selection

6.6.1 Introduction

One of the main differences between existing search engines, such as Google and question answering systems is that an answer to the user’s question and not a list of relevant documents is retrieved. The component ANSWER MATCHING AND SELECTION is responsible for the extraction of answer strings from the set of paragraphs returned by IR.

This is the last stage in the question answering process. As we explained above, we believe that we need rather precise lexico-semantic information such as synonymy. We hope that synonyms will help to overcome the terminological gap, when matching question and answer context.

6.6.2 Description of component

Various syntactic patterns are defined for (exact) answer identification. An important task is therefore to rank potential answers.

The following features are used to determine the score of a short answer $A$ to a question $Q$ extracted from sentence $S$:

**Syntactic Similarity:** The proportion of dependency relations from the question that match with dependency relations in $S$.

**Answer Context:** A score for the syntactic context of $A$ that expresses whether a constituent matching the question type of $Q$ could be found in the right syntactic context in $A$.

**Lexical Overlap:** The proportion of proper names, nouns, verbs, and adjectives from the query which can be found in $S$ and the sentence preceding $S$.

**Frequency:** The frequency of $A$ in all paragraphs returned by IR.

**IR:** The IR score assigned to the paragraph from which $A$ was extracted.

The score for syntactic similarity implements a preference for answers from sentences with a syntactic structure that overlaps with that of the question. Answer context implements a preference for answers that occur in the context of certain terms from the question. Given a question classified as date(Event), for instance, date expressions that occur as a modifier of Event are preferred over date expressions occurring as sisters of Event, which in turn are preferred over dates that have no syntactic relation to Event. The last three features are self-explanatory.
The overall score for an answer is the weighted sum of these features. Weights were determined manually using CLEF data for tuning. The highest scoring answer is returned as the answer.

The use of lexical variants influences both the Syntactic Similarity score and the Lexical Overlap score. Dependency relation triple \((Hd, Rel, Dep)\) matches with \((Hd', Rel, Dep')\), if \(Hd\) and \(Hd'\) are identical or lexical variants according to one of the lexico-semantic sources used. The same holds for \(Dep\) and \(Dep'\).

Lexical knowledge is particularly relevant for the following two special question types:

- WH questions, such as *Which ferry sank southeast of the island Utö?*
- Questions asking for the definition of a person or organisation, i.e. *What is Sabena?*, *Who is Antonio Matarese?*

### 6.6.3 Related work

Pasca (2004) presents methods for acquiring class labels for categorised named entities from unstructured text. The author applies lexico-syntactic extraction patterns based on part-of-speech tags. Patterns were hand-built initially, and extended automatically by scanning the corpus for the pairs of named entities and classes found with the initial patterns. Patterns that occur frequently in matching sentences can be added as additional extraction patterns. Pasca (2004) applies this information to web search for example for processing list-type queries: \(SAS\), \(SPSS\), \(Minitab\) and \(BMDP\) are returned in addition to the top documents for the query *statistical packages*.

The feedback loops in the QA system of Moldovan et al. (2003) (explained in section 6.5.3) are not only applied to query expansion in passage retrieval. At later stages in the question answering process, after the identification of candidate answers, a logic prover verifies the unification of question and logic form of the candidate answer. When the unifications fail, the keywords are expanded with lexico-semantic alternations. This logic proving loop improves the system by 5%.

Pantel and Ravichandran (2004) propose an algorithm that takes a list of semantic classes in the form of clusters of words as input. Labels for these clusters are found by looking at four lexico-syntactic relationships: apposition (*ayatollah Khomeini*), nominal subject (*Khomeini is an ayatollah*), construction with such as (*Ayatollahs such as Khomeini*), and construction with like (*Ayatollahs like Khomeini*). As for the use of categorised named entities, Pantel and Ravichandran (2004) conducted two QA experiments: answering definition questions and
performing information (passage) retrieval. Information retrieval shows small
gains of improvement from using the semantic labels. As for the definition ques-
tions the largest improvements are in the top-5 answers. On the top-1 answers
the system is not able to improve the baselines.

6.6.4 Methodology

We explained in section 6.6.2 that several features are used for ranking candidate
answers.

The application of lexico-semantic knowledge to the features Syntactic Similarity
and Lexical Overlap are most obvious. Instead of using exact matches, words may also match with lexical variations resulting from the three methods
described in section 6.3:

- Nearest neighbours of syntax-based distributional similarity
- Nearest neighbours of alignment-based distributional similarity
- Nearest neighbours of proximity-based distributional similarity

However, each type of lexico-semantic information applied to the component
Answer Matching and Selection will also be used in the IR component. Thus,
the features Frequency and IR are affected as well.

For all methods we selected the top-5 nearest neighbours that had a similarity
score of more than 0.2 as lexical variations. Apart from the three distributional
methods we ran experiments using:

- EuroWordNet
- Categorised named entities (Cat. NEs(2) and Cat. NEs(1) as described
  in section 6.3\textsuperscript{10})

For these two resources we did not set any threshold for inclusion in the list
of lexical variations. We do not have similarity scores for EWN.

The following setting is the same for the lexical variations resulting from
distributional methods and the last two types of lexico-semantic information.

- Lexical variations were added as root forms

**WH questions** We used the categorised named entities to improve the per-
formance of our QA system on WH questions such as:

Which ferry sank southeast of the island Utō?\textsuperscript{10}

\textsuperscript{10}We used Cat. NEs(1) for the WH and definition questions.
6.6. Answer matching and selection

Question analysis and classification tells us that this is a question of type which(ferry). Candidate answers that are selected by our system are: Tallinn, Estonia, Raimo Tiilikainen etc. The QA system uses various strategies to rank potential answers, as we have seen in section 6.6.2. Still, selecting the correct named entity for answers to WH questions poses considerable problems for our system.

To improve the performance of the system on these questions, we incorporated an additional strategy for selecting the correct answer. Potential answers which have been assigned the class corresponding to the question stem (i.e. ferry in this case) are ranked higher than potential answers for which this class label cannot be found in the database of categorised named entities: Tallinn, Estonia, Raimo Tiilikainen etc.. Since Estonia is the only potential answer which is-a ferry, according to our database, this answer is selected.

Definition questions A second question type for which the categorised named entities are relevant are definition questions. The CLEF 2005 QA test set contains no fewer than 60 questions of the form:

What is Sabena?

The named entity Sabena occurs frequently in the corpus, but often with class labels assigned to it, which are not suitable for inclusion in a definition (possibility, partner, company, ...). We already explained that we filtered the list of categorised named entities to get rid of unwanted relatively infrequent labels. Still, there are often several labels available. We selected the most frequent label in case of multiple labels: airline company in this case. Often the class label by itself is not sufficient for an adequate definition. Therefore we expand the class label with modifiers which typically need to be included in a definition.

More in particular, our strategy for answering definition questions consisted of two phases:

- Phase 1: The most frequent class found for a named entity is taken.
- Phase 2: The sentences which mention the named entity and the class are selected, and searched for additional information which might be relevant. Snippets of information that are in an adjectival relation or a prepositional modifier relation to the class label are selected.

For the example above, our system produces Belgian airline company as answer.

However, deciding beforehand what information is relevant is not trivial. As explained, we decided to only expand the label with adjectival and PP modifiers
Chapter 6. Using lexico-semantic knowledge for question answering

<table>
<thead>
<tr>
<th>System</th>
<th># q</th>
<th>MRR</th>
<th>CLEF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>775</td>
<td>.670</td>
<td>.613</td>
</tr>
<tr>
<td>EWN</td>
<td>775</td>
<td>.675</td>
<td>.619</td>
</tr>
<tr>
<td>Syntax</td>
<td>775</td>
<td>.670</td>
<td>.612</td>
</tr>
<tr>
<td>Align</td>
<td>775</td>
<td>.676</td>
<td>.619</td>
</tr>
</tbody>
</table>

Table 6.5: Overall performance (MRR and CLEF-score) of different types of lexico-semantic information on the CLEF ('03, '04, '05) Dutch QA test set.

that are adjacent to the class label in the corresponding sentence. This is the reason for a number of answers being inexact. Given the constituent *the museum Hermitage in St Petersburg*, this strategy fails to include *in St Petersburg*, for instance, because *museum* and *in St Petersburg* are not adjacent. We did not include relative clause modifiers, as these tend to contain information which is not appropriate for a definition. However, in the case of the question, *Who is Iqbal Masih?*, an answer that includes at least the first conjunct of the relative clause of the constituent *twelve year old boy, who fought against child labour and was shot Sunday in his home town Muritke* would have been preferable over just selecting *twelve year old boy*. Similarly, we did not include purpose clauses, which leads the system to respond *large scale American attempt to the question what was the Manhattan project*, instead of *large scale American attempt to develop the first (that is, before the Germans) atomic bomb*.

6.6.5 Evaluation

In this section we evaluate the effect of using the four types of lexico-semantic knowledge for Answer Matching and Selection for QA. We ran an evaluation on the Dutch questions from CLEF ’03, ’04 and ’05 showing the effect of these types of lexico-semantic information.

We calculated for each run the Mean Reciprocal Rank (MRR) and the CLEF score. The MRR score is computed over the five highest ranked answers only, as in section 6.4.5. The CLEF score gives the precision of the first (highest ranked) answer only.

Furthermore, we show for the two special question types, WH questions and definition questions, what the effect is of using categorised named entities.

6.6.6 Results

The results of applying the different types of lexico-semantic information to our QA system Joost are given in Table 6.5. The differences between the systems is very small, EWN and the alignment-based nearest neighbours performing slightly better than the baseline and the syntax-based method. In Table 6.6 the
results are given excluding the questions that are answered by means of off-line table look-up. We had hoped that the exclusion of questions that do not make use of lexico-semantic information in this way would show larger differences. The differences are a little bit more apparent, but still very small.

Although the differences are small they do reflect our intuitions. We predicted that the syntax-based nearest neighbours would not be very suitable for answer matching and selection because of the many co-hyponyms it finds. Remember that we said that we expected tight semantic information to perform best for answer matching and selection. Indeed the tighter semantic lexico-semantic information such as the synonyms from EWN and the nearest neighbours retrieved from the alignment-based method perform best. However, the differences are very small and we might be attaching too much significance to insignificant differences.

The fact that there is such a small difference in performance between the different systems leads us to believe that the number of questions that are affected by the use of lexico-semantic information might be very small. However, from Table 6.7 we can see that this is not the case. When applying lexico-semantic information from EWN, 50 questions in total are affected out of 775. If we subtract the questions handled by off-line QA, we can state that about 10% (50 out of 503) of the questions are affected either positively or negatively by the use of lexico-semantic information. However, unfortunately, the positive and negative effects are equally well represented.

We inspected the expansions in both negatively affected questions and positively affected questions. There appeared to be no pattern that could help us improve the method.

**WH questions and definition questions**

We have applied lexico-semantic information to two special question types: definition questions and wh questions, as explained in section 6.6.4 and section 6.6.4. We used the categorised named entities to improve the handling of these questions.
Table 6.7: Number of questions positively (+) and negatively (-) affected by the use of lexico-semantic information on the CLEF ('03, '04, '05) Dutch QA test set

<table>
<thead>
<tr>
<th>Q-type</th>
<th>Baseline</th>
<th>Improved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#q</td>
<td>MRR</td>
</tr>
<tr>
<td>wh</td>
<td>107</td>
<td>0.40</td>
</tr>
<tr>
<td>Definition</td>
<td>83</td>
<td>0.65</td>
</tr>
<tr>
<td>Person</td>
<td>35</td>
<td>0.74</td>
</tr>
<tr>
<td>Total</td>
<td>775</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 6.8: Overall performance of the baseline and improved QA system on the CLEF ('03, '04, '05) Dutch QA test set.

In table 6.8 the performance of the baseline and improved system is shown. In the first column the question type is given. In the second and fifth column the number of questions classified as being of the corresponding question type is shown. In columns 3 and 6 the corresponding mean reciprocal rank (MRR) score is given. In columns 4 and 7 the corresponding CLEF score is given.

The baseline in these experiments is the Joost QA system without access to lexico-semantic information. wh-questions and definition questions are answered by selecting the most highly ranked answer from the list of relevant paragraphs returned by the IR component. Answers to definition questions are basically selected by means of the same strategy as described for the improved system above, except that answers must now be selected from the documents returned by IR, rather than from sentences known to contain a relevant class label.

Adding categorised named entities as an additional knowledge source for answering wh-questions improves the MRR score of 107 wh-questions by 9% and improves the CLEF score by 11%. Using the same information to provide answers to definition questions improves the MRR score of 83 definition questions by 15% and improves the CLEF score by 20%.

Poor performance of wh-questions affects the performance of person ques-

---

11 Question types not relevant for this experiment are left out.
12 The number of wh-questions is lower in the improved system because due to the use of lexico-semantic information some questions have been classified as FUNCTION or FUNCTION_OF questions, as we have shown in section 6.4.6 and Table 6.2.
6.7. Anaphora resolution for off-line QA

In our experiments we try to improve the technique for off-line answer extraction by applying anaphora resolution. More specifically, we want to extract potential answers from a corpus not only when they are clearly stated with the accompanying named entity in the same sentence, but also when an anaphoric expression is used to refer to an earlier mentioned named entity.\(^\text{13}\)

For instance, consider the question in (11):

\[(11) \text{How old is Ivanisevic?}\]

---

\(^{13}\)In general, work done on anaphora resolution can be classified according to the word class of the anaphora that one tries to resolve: pronouns, proper nouns or definite NPs. We focus on resolving anaphoric definite NPs. We restrict the notion of definite NP to singular NPs modified by the Dutch definite articles 'de' and 'het'.
In order to extract the answer from the text provided in (12), we have to go beyond sentence level.

(12) Todd Martin was the opponent of the quiet Ivanisevic. The American, who defeated the local hero Boris Becker a day earlier, was beaten by the 26-year-old Croatian during the finals of the Grand Slam Cup [...].

Among other things, one must correctly identify Ivanisevic, located in the first sentence, as the denotation of the Croatian, located in the second sentence, in order to extract the correct answer that is stated in the second sentence.

To establish that Ivanisevic is a Croatian we need the information contained in the automatically acquired categorised named entities.

6.7.2 Description of component

As we explained in section 6.4.2, the dependency analysis of a sentence gives rise to a set of dependency relations of the form \( \langle \text{Head/HPos, Rel, Dep/DPos} \rangle \), where \text{Head} is the root form of the head of the relation, and \text{Dep} is the head of the constituent that is the dependent. HPos and DPos are string indices, and Rel is the name of the dependency relation. For instance, the dependency analysis of sentence (13-a) is (13-b).

(13) a. Jacques Chirac was born in Paris.
    b. \[
    \begin{cases}
    \langle \text{was/2, su, Jacques Chirac/1} \rangle, \\
    \langle \text{was/2, vc, born/3} \rangle, \\
    \langle \text{born/3, obj, Jacques Chirac/1} \rangle, \\
    \langle \text{born/3, mod, in/4} \rangle, \\
    \langle \text{in/4, obj, Paris/5} \rangle
    \end{cases}
    \]

We defined dependency patterns to match dependency relations of potential answer phrases. A dependency pattern is a set of (partially underspecified) dependency relations. The following pattern

\[
\begin{cases}
\langle \text{born/B, obj, Name/N} \rangle, \\
\langle \text{born/B, mod, in/I} \rangle, \\
\langle \text{in/I, obj, Location/L} \rangle
\end{cases}
\]

matches with the set in (13-b) and would, among others, instantiate the variable Name with Jacques Chirac and the variable Location with Paris.

\[14\text{We use the CLEF-corpus for our experiments. This corpus consists of newspaper text from 1994 and 1995.}\]
6.7. Anaphora resolution for off-line QA

When a pattern matches, the fact is extracted and stored in a table. In the example above Jacques Chirac and Paris are extracted and the terms together become one entry in the BirthLoc table that holds information about the location in which people are born. If a user should ask our system Where was Jacques Chirac born?, the answer is easily looked up in the table.

6.7.3 Related work

Most work on anaphora resolution involves resolving pronouns. (Mitkov, 1998; Kehler et al., 2004) Relatively high performance is achieved in these experiments: accuracies of 89.7%, 73.4% and 79% respectively. Also for Dutch most work focuses on the resolution of pronominal anaphora. (Op den Akker et al., 2002; Bouma, 2003).

Anaphoric definite NPs are typically harder to resolve. Strube et al. (2002) report that their system performed poorly on definite NPs (precision of 69.26%) and quite well on personal pronouns (precision of 85.81%) and possessive pronouns (precision of 82.11%). The only Dutch work on resolving anaphoric definite NPs was done by Hoste (2005). With respect to common noun co-reference, she obtains precision scores around 47.5% compared to precision scores around 65% for pronouns.

It is known that the resolution of full NPs requires a large and diverse amount of world knowledge. Many systems that deal with resolution of full NPs (Harabagiu et al., 2001; Ng and Cardie, 2002) use manually constructed lexico-semantic resources such as WordNet (Fellbaum, 1998). But as Markert and Nissim (2005) explain, such resources are often not sufficient. The authors describe several problems related to the use of such lexico-semantic knowledge resources for anaphora resolutions, of which two relate strongly to our work.

One problem they describe is the lack of knowledge in such knowledge resources. Though carefully built, the lack of knowledge is often severe. For the present application this lack of knowledge is even more severe. Firstly, we are working on Dutch and the resources for languages other than English have even less coverage. Secondly the type of knowledge that we use, categorised named entities, is typically not very well represented in lexico-semantic knowledge bases. The fact that Guus Hiddink is a national team coach is not available in EWN.

A second problem is the fact that relations between words needed for coreference resolution for full NPs are often very context dependent. These relations are often not found in ontologies. To stay with our previous example Guus Hiddink was the national team coach of the Netherlands in 1995. The document collection we are using for the CLEF track is from 1994-1995. We will hence
often find the categorized named entity Guus Hiddink is a national team coach. Corpus-based methods are more suitable for finding these context-dependent relations.

Because of these problems, a number of researchers have used enhanced knowledge bases for anaphora resolution (Poesio et al., 2002; Markert et al., 2003). In these works the knowledge bases are enhanced (semi)automatically with knowledge from corpora. Markert and Nissim (2005) have extended the corpus-based approach by an approach that exclusively extracts the required knowledge from the Web using shallow lexico-syntactic patterns. Our approach is comparable to their approach in that we do not use any hand-crafted lexical knowledge base either.

Our approach differs in that we do not use the Web, but a large corpus to extract lexico-semantic knowledge. Secondly, we apply syntactic patterns to extract knowledge from parsed text. Thirdly, we focus on selecting named-entities as antecedents, hence the knowledge we extract from our corpus is aimed at named entities. Whereas Markert and Nissim (2005) resolve named entities to one of the following three classes (person, location, organisation), we leave the named entities as they are and construct a knowledge base especially for named entities. The web-based method does not outperform the WordNet-based method in their experiments, but results are comparable. They reach a precision of 0.751 when number-checking is taken into account for the WordNet-based method.

### 6.7.4 Methodology

In this experiment we focused on several question types that we expect will benefit from anaphora resolution. We have applied information contained in the list of categorized named entities (Cat. NEs(1), as described in section 6.3).

For the question type **Age** we extracted a person’s name and his or her age. We extracted names of persons along with the date and location of their birth and stored them in the respective tables (Birth\_date, Birth\_loc). We did the same for the location and date of death of people and the age they have reached (Died\_date, Died\_loc, Died\_age). We also extracted ways in which people have died (Died\_how). In the Inhabitants table we stored names of locations and the accompanying number of inhabitants. Finally, for the Founders table we extracted who founded what at what time. For our experiments we adjusted the patterns. Instead of looking for a named entity we looked for a definite NP. For instance, the new pattern for the example in section 6.7.2 becomes as follows:
6.7. Anaphora resolution for off-line QA

It will match the dependency relations of a sentence, such as (14).

\[(\text{born/B, obj, DefNoun/N}),\]
\[(\text{born/B, mod, in/I}),\]
\[(\text{in/I, obj, Location/L})\]

(14) The president of France was born in Paris.

In the case of the Died\_loc, the Birth\_loc and the Founders table the regular patterns try to fill two slots with two different named entities. In these cases we only replaced the slot which should be filled with a person's name. So, anaphora resolution will only be carried out on one element of the extracted fact: the slot that originally required a person's name.

To be able to extract potential answers from a corpus not only when they are clearly stated with the accompanying named entity in the same sentence, but also when an anaphoric expression is used to refer to an earlier mentioned named entity, we have to apply anaphora resolution. More specifically we have to resolve the definite NPs and find the named entities they refer to.

Our first strategy (Mur and Van der Plas, 2007) for doing that is as follows: We scan the left context of the definite NP for named entities from right to left (i.e. the closest named entity is selected first). For each named entity we encounter, we check whether it is in an is-a relation with the definite NP according to the list of categorised named entities. If so, the named entity is selected as the antecedent of the NP. As long as no suitable named entity is found we select the next named entity and so on, until we reach the beginning of the document. We have limited our search to the current document. If no suitable named entity is found, i.e., no named entity is found that is in an is-a relation with the definite NP, no fact is extracted.

After having resolved the NP, the fact is added to the facts table. In order to explain our strategy for resolving definite NPs we will apply it to the example from the introduction:

(15) Todd Martin was the opponent of the quiet Ivanisevic in December 1995. Todd Martin, who defeated the local hero Boris Becker a day earlier, was beaten by the 26-year old Croatian during the finals of the Grand Slam Cup in 1995 [...].

In (15), the left context of the NP the 26-year old Croatian is scanned from
right to left. The named entities Boris Becker and Todd Martin are selected before the correct antecedent Ivanisevic. The fact that neither Boris Becker nor Todd Martin is found in an is-a relation with Croatian sets them aside as unsuitable candidates. Then Ivanisevic is selected and this candidate is found to be in an is-a relation with Croatian, so Ivanisevic is taken as the antecedent of Croatian. The fact Ivanisevic, 26-year old is added to the Age table.

In Van der Plas et al. (2008a) we chose to use a fallback in case none of the named entities in the document were in an is-a relation with the NP. In that case we extracted the named entity in the previous sentence that is nearest to the anaphoric expression. If no named entity is present in the previous sentence, the NP is not resolved. We will present results for both strategies in the Results section (6.7.6).

6.7.5 Evaluation

The aim of the evaluation is to determine whether anaphora resolution using categorised named entities helps to acquire more facts without hurting the quality of the table.

We developed a simple baseline that always selects the antecedent closest to the anaphor. We chose to use this method as our baseline in spite of experiments done by Markert and Nissim (2005) showing that recency in general is not a good predictor for selecting an antecedent for definite NPs. The reason for this decision lies in the fact that the baseline method that performed best in their experiments, e.g., a string-based approach is not applicable in our setting. An anaphoric definite NP and a named entity have very little string overlap e.g., Van Gogh and painter.

For our evaluation we compared the precision of three types of tables.

- The original tables, i.e., acquired without using anaphora resolution: Original
- The facts added by extracting the most recent named entity as the antecedent of the definite NP: Baseline
- The facts added by using categorised named entities for anaphora resolution: Cat. NEs (1)

Note that for estimating the precision scores for the baseline and the instance method we only looked at the added facts.

We manually evaluated 1% of the largest tables. For each of the smaller tables we chose to evaluate 20 facts. This was always over 1% of the total number of facts of that table.

We evaluated the facts on two criteria:
6.7. Anaphora resolution for off-line QA

- Correctness of the established coreference
- Correctness of the fact

In the evaluation of the second strategy that uses the fallback we used an evaluation strategy that concentrates on the differences between the two methods.

6.7.6 Results

In this section we will present results from two experiments presented in previous work (Mur and Van der Plas, 2007; Van der Plas et al., 2008).

<table>
<thead>
<tr>
<th>Question type</th>
<th>Original types</th>
<th>Original tokens</th>
<th>Baseline types</th>
<th>Baseline tokens</th>
<th>Cat. NEs types</th>
<th>Cat. NEs tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>18,518</td>
<td>22,140</td>
<td>22,798</td>
<td>27,786</td>
<td>19,119</td>
<td>23,350</td>
</tr>
<tr>
<td>Born_date</td>
<td>1,988</td>
<td>2,353</td>
<td>2,162</td>
<td>2,533</td>
<td>1,997</td>
<td>2,365</td>
</tr>
<tr>
<td>Born_loc</td>
<td>876</td>
<td>934</td>
<td>1,071</td>
<td>1,141</td>
<td>908</td>
<td>973</td>
</tr>
<tr>
<td>Died_age</td>
<td>832</td>
<td>1,124</td>
<td>889</td>
<td>1,186</td>
<td>841</td>
<td>1,135</td>
</tr>
<tr>
<td>Died_loc</td>
<td>581</td>
<td>661</td>
<td>615</td>
<td>697</td>
<td>583</td>
<td>665</td>
</tr>
<tr>
<td>Died_date</td>
<td>542</td>
<td>580</td>
<td>707</td>
<td>758</td>
<td>553</td>
<td>596</td>
</tr>
<tr>
<td>Inhabitants</td>
<td>635</td>
<td>705</td>
<td>900</td>
<td>1,002</td>
<td>729</td>
<td>817</td>
</tr>
<tr>
<td>Founded</td>
<td>951</td>
<td>1,018</td>
<td>969</td>
<td>1,036</td>
<td>951</td>
<td>1,018</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>24,923</strong></td>
<td><strong>29,515</strong></td>
<td><strong>30,111</strong></td>
<td><strong>36,139</strong></td>
<td><strong>25,681</strong></td>
<td><strong>30,919</strong></td>
</tr>
</tbody>
</table>

Table 6.9: Number of facts extracted

<table>
<thead>
<tr>
<th>Question type</th>
<th>Original</th>
<th>Baseline</th>
<th>Cat. NEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>72%</td>
<td>39%</td>
<td>79%</td>
</tr>
<tr>
<td>Born_date</td>
<td>100%</td>
<td>15%</td>
<td>50%</td>
</tr>
<tr>
<td>Born_loc</td>
<td>90%</td>
<td>55%</td>
<td>90%</td>
</tr>
<tr>
<td>Died_age</td>
<td>95%</td>
<td>65%</td>
<td>100%</td>
</tr>
<tr>
<td>Died_loc</td>
<td>90%</td>
<td>30%</td>
<td>100%</td>
</tr>
<tr>
<td>Died_date</td>
<td>65%</td>
<td>20%</td>
<td>77%</td>
</tr>
<tr>
<td>Inhabitants</td>
<td>85%</td>
<td>35%</td>
<td>60%</td>
</tr>
<tr>
<td>Founded</td>
<td>90%</td>
<td>24%</td>
<td>-</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>80%</td>
<td>36%</td>
<td>79%</td>
</tr>
</tbody>
</table>

Table 6.10: Estimated precision

Table 6.9 shows how many facts were extracted per table for the original method, the baseline, and the method using categorised named entities. The second and third column show the number of types and tokens respectively when applying the method without anaphora resolution. The fourth and fifth column show the number of types and tokens respectively when adding the facts extracted by the baseline method to the original table. The sixth and seventh
Chapter 6. Using lexico-semantic knowledge for question answering

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Cat. NEs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+ fall back</td>
</tr>
<tr>
<td>Age</td>
<td>20,229</td>
<td>24,917</td>
</tr>
<tr>
<td>Born.date</td>
<td>2,297</td>
<td>2,395</td>
</tr>
<tr>
<td>Born.loc</td>
<td>795</td>
<td>948</td>
</tr>
<tr>
<td>Died.age</td>
<td>923</td>
<td>966</td>
</tr>
<tr>
<td>Died.loc</td>
<td>720</td>
<td>744</td>
</tr>
<tr>
<td>Died.date</td>
<td>1,011</td>
<td>1,204</td>
</tr>
<tr>
<td>Died.how</td>
<td>1,834</td>
<td>2,336</td>
</tr>
<tr>
<td>Total</td>
<td>27,809</td>
<td>33,510</td>
</tr>
</tbody>
</table>

Table 6.11: Number of facts found for the different tables for relation extraction

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>168</td>
<td>128</td>
</tr>
<tr>
<td>Increase freq.</td>
<td>95</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>263</td>
<td>137</td>
</tr>
</tbody>
</table>

Table 6.12: Distribution of facts that differ between original tables and tables that use categorised named entities + fallback for sample of 400 differences

columns show the number of types and tokens respectively when adding the facts extracted by the method using categorised named entities (Cat. NEs (1)) to the original table. The total number of new facts added (types) was 5,088 for the baseline and 778 for the categorised named entity method. These numbers seem to speak in favour of the baseline method, but the precision scores point in the opposite direction.

In table 6.10 the estimated precision per table for each of the three methods is given. The average precision scores based on the set of manually evaluated samples are 80% for the original tables 36% for the baseline and 79% for the method using categorised named entities. The estimated precision of the tables acquired by applying anaphora resolution using the categorised named entities is only 1% lower than the estimated precision of the original tables constructed without using anaphora resolution. The estimated precision of the baseline is less than half of the estimated precision reached by the other two methods.

However, the number of facts extracted by the method using categorised named entities is rather small. Only 758 facts (types) are extracted with this method in addition to the 24,923 available in the original tables. We have therefore decided in Van der Plas et al (2008a) to use a fall back strategy. It consists of selecting the named entity in the previous sentence, in case there is no is-a relation between the NP and any of the named entities in the paragraph. In table 6.11 we can see that the number of facts added (types) is indeed larger than when no fallback is used. There are 5,701 facts added in total.
We extracted all differences between entries (types) in the original table and the table that uses anaphora resolution. These differences can be either new facts or increases in frequency. From these differences we randomly extracted 400 entries. Two of the authors of Van der Plas et al. (2008a) determined the correctness of the found facts in both tables. The results are given in table 6.12.\footnote{In Van der Plas et al. (2008a) we have included results for the question type Died how, in Mur and Van der Plas (2007) we have not. However, in Mur and Van der Plas (2007) we have included results for the question types Founded and Inhabitants, which we did not include in Van der Plas et al. (2008a). This is why there are results represented for different question types in Table 6.9 and Table 6.12.}

A large number of facts (104 from 400) show a rise in frequency and 95 of these 104 are correct facts. This is a positive result with regard to the reliability of the table. The precision of the facts, however, is not very encouraging. Overall, 263 (66\%) of the 400 facts are correct. It seems that the fall back method hurts precision a lot.

With regard to experiments on off-line QA we were not able to show that using lexico-semantic driven anaphora resolution for relation extraction improves the performance of the system on the CLEF test set. We believe that this is due to the fact that the test set contains only 19 questions with a question type for which anaphora resolution potentially could make a difference, i.e., questions that were of one of the question types (see table 6.11) for which the relation extraction module using anaphora resolution provides answers. Another reason might be the earlier mentioned presence of back formulations in the question collection. In the case of back formulations the information is usually stated within one sentence, and hence there is no use for anaphora resolution.

### 6.7.7 Conclusion

In this section we have applied categorised named entities to the task of anaphora resolution for definitive NPs in the component of off-line QA. We have given results for two strategies. The first strategy uses the categorised named entities to select a suitable antecedent within the current document. If no suitable antecedent is found, no fact is extracted. The second strategy uses a fallback in case no suitable antecedent is found according to the list of categorised named entities. We can conclude from the experiments that the first strategy is a very precise strategy. The estimated precision of the tables built using anaphora resolution with categorised named entities is only 1\% lower (79\%) than the precision of the original tables using no anaphora resolution (80\%). The number of facts added by this method is modest: 758 compared to 5,188 when using the fallback method. However, the estimated precision of the facts extracted by the fallback method amounts to 66\% only.
We were not able to show that the larger tables built by using the second strategy for anaphora resolution improved the QA system. We believe that this is due to the fact that only 19 questions out of 775 were of the question type for which anaphora resolution could make a difference. Furthermore, the fact that some questions seems to be back formulations of sentences stated literally in the corpus makes anaphora resolution less relevant.

6.8 Conclusions

In this chapter we have applied several lexico-semantic resources to the task of open-domain question answering (QA). We have applied the information to several components of the QA system. We will briefly summarise the main outcomes per component.

Question analysis is improved by expanding the list of functions from EWN semi-automatically with nearest neighbours from the syntax-based method. A total of 88 questions are classified as function or function of questions, whereas only 71 questions were selected on the basis of the EWN function list. Adding a question class for functions and function of results in a shift from person and WH questions to function and function of questions. A shift that is in general beneficial for the performance of the QA system.

Query expansion for passage retrieval with synonyms to overcome the terminological gap is not very fruitful. We believe that the noise introduced by ambiguity of the query terms is more important than the positive effect of adding lexical variants. However, adding extra information with regard to the subject field of the query instead of synonyms, related words bridging the knowledge gap, proved slightly beneficial. The proximity-based expansions augment the MRR scores with 1.5%. Most successful are the categorised named entities. These expansions were able to augment the MRR scores with nearly 3.5%. Also, we conclude from the experiments that corpus-based resources provide more relevant information than hand-built resources, if the corpus used to retrieve the information from is the same as the corpus used in the QA task. This is especially true for proper names.

With respect to the component answer matching and selection we can conclude that the most positive results stem from applying categorised named entities to particular question types: WH-questions and definition questions. The CLEF scores are improved by 11% and 20%, respectively. The improvement on the WH-questions has a positive effect on person questions as well. The CLEF score for person questions is improved by 20%. The use of lexico-semantic information in the form of synonyms to the answer matching strategy overall does not result in large improvements.
6.8. Conclusions

Lastly we have applied categorised named entities to the task of anaphora resolution for definitive NPs in the component of off-line QA. We have provided results for two strategies. The first strategy uses the categorised named entities to select a suitable antecedent within the current document. The second strategy uses a fallback in case no suitable antecedent is found according to the list of categorised named entities. We can conclude from the experiments that the first strategy is a very precise strategy, but the number of facts added by this method is modest: 758 compared to 5,188 when using the fallback method. However, the estimated precision of the facts extracted by the fallback method amounts to 66% only.

Now that we have discussed the results per module, we would like to give a short overview of the usefulness of the different types of lexico-semantic information.

It seems that the most fruitful type of lexico-semantic information are the categorised named entities. They proved beneficial for the passage retrieval component, for use in definition and wh questions, and for the task of anaphora resolution for definitive NPs in the component of off-line QA. It is a known fact that named entities are important for a task such as open-domain QA. People often ask questions about persons or organisations, such as *Who is Johannes Vermeer?*. They also often ask questions that have a named entity as an answer, such as *Who is the tennis player who got stabbed in the back?*. On the other hand, we should not forget that the questions from the CLEF testset are not questions from real users. The fact that there is so much emphasis on proper names might be due to the setup of the cross language evaluation forum.

The syntax-based nearest neighbours proved beneficial, when used in a semi-automatic way to expand a list of co-hyponyms (functions people have) from EuroWordNet.

The proximity-based nearest neighbours proved slightly beneficial for query expansion in passage retrieval, although the categorised named entities outperform the proximity-based nearest neighbours in this task.

We believe that these experiments are just a first step towards trying to make lexico-semantic information useful for QA. For example, we would like to try using techniques for query expansion that filter out expansions that result from inappropriate word senses. We tried a simple overlap method and would like to try to expand this method. Also, we would like to investigate how many lexical variation can be found between the question and the sentences that hold the answer in the CLEF test sets and for questions from users. If there is little variation by consequence little can be gained by using query expansion methods that try to bridge the terminological gap.