Chapter 3

Extraction Based on Dependency Relations

3.1 Introduction

The results of the experiments in chapter 2 showed that surface patterns are often not flexible enough to extract the information needed to answer a question. The sentences in (1), for instance, all contain information about organisations and their founders.

(1) a. Minderop richtte de Tros op toen [...]  
   English: Minderop founded Tros when [...]  

   b. Op last van generaal De Gaulle in Londen richtte verzetsheld Jean Moulin in mei 1943 de Conseil National de la Résistance (CNR) op.  
   English: Following orders of general De Gaulle, resistance hero Jean Moulin founded in May 1943 the Conseil National de la Résistance (CNR).  

   c. Kasparov heeft een nieuwe Russische Schaakbond opgericht [...]  
   English: Kasparov has founded a new Russian Chess Union [...]  

   d. [...] toen de Generale Bank bekend maakte met de Belgische Post een “postbank” op te richten.  
   English: [... ] when the General Bank announced to found a “postal bank” with the Belgian Mail.

The verb oprichten ‘to found’ can take on a wide variety of forms (active, with the particle op split from the root, as participle, and infinitive). Both the founder and the organisation can be the first constituent in the sentence, as well as other elements including adjuncts. In control constructions the founder may be the subject of a governing clause. In all cases, modifiers may intervene between the relevant constituents. Such variation is almost impossible to capture accurately using surface strings, whereas dependency relations can exploit the fact that in all cases the organisation and its founder can be identified as the object and subject of the verb with the root
form *richt-op*. In general, dependency relations allow patterns to be stated which are hard to capture using regular expressions.

Still, although dependency relations eliminate many sources of variation that systems based on surface strings have to deal with, it is also true that the same semantic relation can sometimes be expressed by several dependency relation patterns. For example the subject of an active sentence (2-a) may be expressed as a PP-modifier headed by *door* ‘by’ in the passive (2-b).

(2) a. Martin Batenburg *richtte* op 1 december Het Algemeen Ouderen Verbond
   English: Martin Batenburg founded The General Pensioners Union on December 1st.

   b. Het Algemeen Ouderen Verbond *is* op 1 december *opgericht* door Martin Batenburg.
   English: The General Pensioners Union was founded on December 1st by Martin Batenburg.

Lin and Pantel (2002) show how dependency paths expressing the same semantic relation can be acquired from a corpus automatically. Rinaldi et al. (2003) argue that such equivalences, or paraphrases, can be especially useful for QA in technical domains.

Another problem arises with questions like (3-a) and (3-b):

(3) a. Wie is de Duitse minister van Economische Zaken?
   English: Who is the German minister of Economy?

   b. Wie was president van de VS in de Tweede Wereldoorlog?
   English: Who was president of the US during the second World War?

These are hard to handle using off-line methods. Question (3-a) and (3-b) would be classified, by question analysis, as *function(minister,duits)* and *function(president,vs)* respectively, and thus, in principle can be answered by consulting the function-table. However, this ignores the modifiers *van Economische Zaken* and *in de Tweede Wereldoorlog*, which, in this case, are crucial for finding the correct answer.

In this chapter we investigate more flexible techniques for answer extraction than those applied in chapter 2. We present a comparison between a dependency-pattern-based approach and a surface-pattern-based approach on a set of CLEF questions.
3.2. Answer Extraction

We compare the amount of information extracted and number of correctly answered questions for both techniques. Furthermore, syntactic variation is accounted for by formulating equivalence relations over dependency patterns. More on this technique is explained in section 3.2.2.1. In addition, we introduce the d-score in section 3.2.2.2, which computes the extent to which the dependency structure of question and answer match so as to take into account crucial modifiers that otherwise would be ignored. We show that both the use of dependency matching in general, as well as the addition of equivalence rules and the addition of the d-score improve the performance of QA.

The remainder of the chapter is organised as follows: In section 2 we provide details on the extraction methods used, we describe the equivalence rules applied, and we introduce the d-score. Section 3 contains a description of our experiments and results. A discussion on the results is given in section 4. We conclude in section 5.

Earlier versions of work discussed in this chapter have appeared as Jijkoun et al. (2004) and Bouma et al. (2005).

3.2 Answer Extraction

Clearly, the performance of an information extraction module depends on the set of language phenomena or patterns covered, but this relation is not straightforward: having more patterns allows one to find more information, and thus increases recall, but it may also introduce additional noise that hurts precision. Since in our experiments we aimed at comparing extraction modules based on surface patterns vs. syntactic patterns, we first tried to define patterns that are technically good and which cover at least the most intuitive language phenomena per category. Second we tried to keep the two modules parallel in terms of the phenomena covered. We also kept the number of patterns the same in the two modules.

We selected six question types for which we will define patterns, namely capital (e.g. France - Paris), currency (e.g. US - dollar), date-of-birth (e.g. Walt Disney - 1901; Vincent van Gogh - 30 March 1853), founder (e.g. Henry Dunant - Red cross - 1864), function (e.g. Clinton - president - US\(^1\)), and location-of-birth (e.g. Mozart - Salzburg ). Questions about one of these six relations cover a variety of answer types, ranging from person names and location names to dates and currencies.

In order to use dependency patterns the complete corpus was parsed by Alpino (Van Noord, 2006). Besides dependency relations the annotation also included root-
form information, POS-information and named-entity information. Apart from the
information about dependency relations, all information is also used in the surface-
pattern-based extraction method. More information about Alpino is given in chapter
2, section 2.1.2.

To further improve the performance of the off-line answer extraction module we
implemented equivalence rules and used the d-score for evaluating candidate answers,
both shortly introduced in the previous section. Below we first describe the two
extraction methods, then we provide more details about the equivalence rules and the
d-score.

3.2.1 Extraction with Surface Patterns

To extract information about the above-mentioned relations, we extended the set of
surface patterns used for the experiments in chapter 2. We defined a total of 21
patterns, which cover common structures in terms of the information sought. Six rep-
resentative examples (one for each relation) are given in figure 3.1. For the complete
set of surface patterns see appendix A where they are listed alongside the dependency
patterns.

- /[ADJC] [hoofdstad] [,] ? [NAME]/
  Franse hoofdstad Parijs ‘French capital Paris’

- /[COIN] [in] [COUNTRY] [BE] [TERM] [CURRENCY]/
  De munteenheid in Hongarije is de forint ‘The currency in Hungary is the forint’

- /[NAMEP] [(] [YEARB – YEARD)]/
  Jan (1900-1985) ‘John (1900-1985)’

- /[NAME] [VERBp] [TERM]{2} [NAMEO] ([PREP] [DATE])?
  Chirac richtte de RPR op […] ‘Chirac founded the RPR’

- /[ADJC] [FUNCTION] [,] ? [NAMEP]/
  Franse president Sarkozy ‘French president Sarkozy’

- /[NAMEP] [,] [geboren] [in] [NAME]/
  Jan, geboren in Amsterdam ‘John, born in Amsterdam’

Figure 3.1: Sample of surface patterns

In these patterns, all matching terms are between brackets. The terms in bold are
the ones matching with phrases we would like to extract. NAME is a phrase that is
tagged as name by the POS-tagger. $NAME_P$ and $NAME_O$ are phrases tagged as person name and organisation name by the Named Entity tagger respectively. $DATE$ is an expression of time identified by the POS-tagger, $YEAR$ should match with a number of four digits (the subscriptions B and D are used to distinguish between the birth year and year of death). $ADJ_C$, $COUNTRY$, $CURRENCY$, and $FUNCTION$ are all phrases from lists. $BE$ is a form of the verb to be. $VERB_F$ is one of the founding verbs ‘stichten’ (found), ‘oprichten’ (set up) or ‘instellen’ (establish). $PREP$ is one of the preposition ‘in’ (in) or ‘op’ (at). $COIN$ matches either the term ‘munt’ (coinage) or ‘munteenheid’ (currency). $TERM$ can match with an arbitrary term. Finally, we used a few quantifiers. The question mark indicates zero or one occurrences of the preceding element. \{n\} means between zero and n occurrences of the preceding element.

The entries on the $ADJ_C$, $COUNTRY$, and $CURRENCY$ lists are manually collected from databases found on the Internet. The lists contain 90, 204 and 1031 entries respectively. The list with function terms was partly taken from the Dutch part of the EuroWordNet (Vossen, 1998) consisting of all words under the node leider ‘leader’, 255 in total. Since the coverage of the Dutch EWN is far from complete, Van der Plas and Bouma (2005) employed a technique based on distributional similarity to extend the list automatically. We used the extended list for our patterns.

### 3.2.2 Extraction with Syntactic Patterns

To use the syntactic structure of sentences for answer extraction, the collections were parsed with Alpino. Figure 3.2 lists dependency patterns which form the counterparts of the surface patterns in figure 3.1. The dependency relations are given in the form of triples: \langle Head/HPos, Rel, Dep/DPos \rangle, where Head is the root form of the head of the relation, and 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. The entity classes used are the same as for the surface patterns. As for the labels of the dependency relations, subj denotes the subject relation, obj1 is the direct object relation, mod stands for the modifier relation, app is the apposition relation, and predc denotes the predicate-complement relation.

The complete set of dependency patterns is found in appendix A.
Chapter 3. Extraction based on dependency relations

3.2.2.1 Equivalence rules

Equivalences can be defined to account for the fact that in some cases we want a pattern to match a set dependency relations that differs from it, but nevertheless expresses the same semantic relation. For instance, the subject of an active sentence may be expressed as a PP-modifier headed by *door (by)* in the passive:

(4) a. Zimbabwe verleende asiel aan Mengistu.  
    English: Zimbabwe granted Mengistu asylum.

   b. Aan Mengistu werd asiel verleend door Zimbabwe.  
    English: Mengistu was given asylum by Zimbabwe.

The following equivalence rule accounts for this:

\[
\text{equiv} \left( \{ \langle \text{word/W, vc, Vb/V} \rangle, \langle \text{Vb/V, subj, Su/S} \rangle \} \right), \left\{ \langle \text{Vb/V, mod, door/D} \rangle, \langle \text{door/D, obj1, Su/S} \rangle \right\}. \]

Figure 3.2: Sample of dependency patterns
3.2. Answer Extraction

Here, the verb word is the root form of the passive auxiliary, which takes a verbal complement \( vc \) headed by the verb \( Vb \).

We have implemented 13 additional equivalence rules to account for, among others, coordination, relative clauses, and possessive relations expressed by the verb \textit{hebben} (to have). Here are some examples:

(5)  
\begin{enumerate}
  \item a. de bondscoach van Noorwegen, Egil Olsen \iff\ Egil Olsen, de bondscoach van Noorwegen
      English: the coach of Norway, Egil Olsen \iff\ Egil Olsen, the coach of Norway
  \item b. Australië’s staatshoofd \iff\ staatshoofd van Australië
      Australia’s head of state \iff\ head of state of Australia
  \item c. president van Rusland, Jeltsin \iff\ Jeltsin is president van Rusland
      English: president of Russia, Jeltsin \iff\ Jeltsin is president of Russia
  \item d. Moskou heeft 9 miljoen inwoners \iff\ de 9 miljoen inwoners van Moskou
      English: Moscow has 9 million inhabitants \iff\ the 9 million inhabitants of Moscow
  \item e. Swissair en Austrian Airlines hebben vluchten naar Kroatië \iff\ Swissair heeft vluchten naar Kroatië
      English: Swissair and Austrian Airlines have flights into Croatia \iff\ Swissair has flights into Croatia
  \item f. Ulbricht liet de Berlijnse Muur bouwen \iff\ Ulbricht, die de Berlijnse Muur liet bouwen
      Ulbricht had the Berlin Wall be built \iff\ Ulbricht, who had the Berlin Wall be built
\end{enumerate}

The equivalence rules we have implemented express linguistic equivalences, and thus are both general and domain independent.

Once we define a pattern to extract the country and its capital from (6-a), the equivalence rules illustrated in (5-a), (5-b), and (5-c) can be used to match this single pattern against the alternative formulations in (6-b)-(6-d) as well.

(6)  
\begin{enumerate}
  \item a. de hoofdstad van Afghanistan, Kabul
      English: the capital of Afghanistan, Kabul
  \item b. Kabul, de hoofdstad van Afghanistan
      English: Kabul, the capital of Afghanistan
  \item c. Afghanistan's hoofdstad, Kabul
\end{enumerate}
English: Afghanistan’s capital, Kabul

d. Kabul is de hoofdstad van Afghanistan

English: Kabul is the capital of Afghanistan

3.2.2.2 D-score

The d-score introduced in Bouma et al. (2005) between the question and the sentence on which a table entry is based, can be used as an additional factor (in conjunction with frequency) in determining whether a table answer is correct. The d-score computes to what extent the dependency structure of question and answer match. To this end, the set of dependency relations of the question is turned into a pattern \( Q \), by removing the dependency relations for the question word, and then substituting variables for the string positions. We then want to calculate how many dependency relations in the question pattern \( Q \) also occur in the set of dependency relations of the answer \( A \). In other words, we want to know the cardinality of the largest subset \( Q' \) of \( Q \) such that all relations in \( Q' \) match with relations in \( A \) (\( \text{match}(Q', A) \) holds). To obtain the d-score we divide by the cardinality of the set of dependency relations in the question pattern \( |Q| \):

\[
\text{d-score}(Q, A) = \frac{\arg \max_{Q'} |\{Q' | Q' \subset Q \land \text{match}(Q', A)\}|}{|Q|}
\]

For instance, for question (7) classified as \text{function(minister,duits)}\), there are several candidate answers, some of which are Klaus Kinkel (frequency 54), Theo Waigel (frequency 36), Volker Rühe (frequency 15), and Günter Rexrodt (frequency 11).

(7) Wie is de Duitse minister van Economische Zaken?

English: Who is the German minister of Economy?

In this case, using frequency only to determine the correct answer, would give the wrong result (Klaus Kinkel, he was the German minister of Foreign Affairs), whereas a score that combines frequency and d-score (based on (8), on which one of the table entries was based) returns the correct answer: Günter Rexrodt.

(8) De Duitse minister van Economische Zaken, Günter Rexrodt, verwelkomde het rapport.

English: The German minister of economy, Günter Rexrodt, welcomed the report.
Note that dependency matching is considerably more subtle than keyword matching. A case in point are Q/A-pairs such as the following:

(9)  
\begin{enumerate}
  \item Wie is voorzitter van het Europese Parlement?
        English: Who is chair of the European Parliament?
  \item Karin Junkers (SPD), lid van het Europese Parlement en voorzitter van de vereniging van sociaal-democratische vrouwen in Europa [...] 
        English: Karin Junkers (SPD), member of the European Parliament and chair of the society of social-democrat women in Europe [...] 
\end{enumerate}

Here, (9-b) does not contain the correct answer in spite of the fact that it contains all keywords from the question. In fact, even most of the dependency relations of the question are present in the answer, but crucially, there is no substitution for \( \tilde{W} \) that would satisfy:

\[
\text{match}\left\{ \left\{ \text{voorzitter/V,mod,van/W}, \right\}, \left\{ \text{van/W,subj1,parlement/X} \right\} \right\}, Q
\]

### 3.3 Experiments

In this section we describe the experiments performed. First we present results for the extraction task, then we provide details about the question answering task. The discussion of the results follows in section 3.4.

#### 3.3.1 Extraction task

Three separate extraction experiments were carried out, one using surface patterns, one using dependency patterns and one using dependency patterns with the addition of equivalence rules.

We defined patterns for the six question types introduced earlier. The surface patterns and the corresponding dependency patterns are listed in appendix A. The equivalence rules have been described in section 3.2.2.1.

Facts are extracted by matching the patterns to sentences in the CLEF corpus. The CLEF corpus is described in the previous chapter; recall that it was completely parsed by Alpino.

The numbers of extracted facts per extraction method are listed in table 3.1. From each set of extracted facts we took a random sample of one hundred facts and evaluated
Chapter 3. **Extraction based on dependency relations**

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Surface</th>
<th>Dependency</th>
<th>Dependency + eq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>capital</td>
<td>2105 (92%)</td>
<td>1959 (99%)</td>
<td>2078 (96%)</td>
</tr>
<tr>
<td>currency</td>
<td>2875 (99%)</td>
<td>2844 (98%)</td>
<td>2880 (96%)</td>
</tr>
<tr>
<td>date-of-birth</td>
<td>2024 (92%)</td>
<td>1424 (95%)</td>
<td>2300 (94%)</td>
</tr>
<tr>
<td>founder</td>
<td>326 (68%)</td>
<td>517 (74%)</td>
<td>1185 (75%)</td>
</tr>
<tr>
<td>function</td>
<td>38423 (72%)</td>
<td>43596 (89%)</td>
<td>50643 (78%)</td>
</tr>
<tr>
<td>location-of-birth</td>
<td>84 (99%)</td>
<td>479 (98%)</td>
<td>744 (94%)</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td>45837 (79%)</td>
<td>50819 (93%)</td>
<td>59830 (80%)</td>
</tr>
</tbody>
</table>

Table 3.1: Extraction results with estimated precision scores between brackets

them manually as we did in chapter 2.\(^2\) Between brackets are the estimated precision scores based on the evaluation of these random samples.

The most salient result is the high precision score overall. This outcome supports findings of related work discussed in chapter 1, section 1.2.1 (Tjong Kim Sang et al., 2005). Only for the founder facts extracted using surface patterns the precision is a bit lower, but this is not surprising given the wide variety of forms in which these facts can occur, as illustrated in the beginning of this chapter in example (1).

Furthermore, the precision scores for the results obtained with the dependency based method seem to be higher than the precision scores for the results obtained with the surface based method. Using equivalence rules tempered the scores again a bit, but it remained on the same level as the precision score for surface patterns.

Calculating the confidence intervals at level 95% for these results we get (0.71;0.87) for the results obtained with the surface based method, (0.87;0.98) for the results obtained with the dependency based method and (0.72;0.88) for the results obtained by adding equivalence rules.

Looking at the total amount of facts extracted we can see that using dependency patterns we extract around 5000 (ca. 11%) facts more compared to using surface patterns. Adding equivalence rules increased the number with more than 9000 (18%) facts.

Although it is not possible to compute the recall score, since the total number of correct facts in the corpus is not known, we can say something about the relative recall. 45837 facts for the surface based method with precision 79% means there must be at least 36211 positive instances, while 50819 facts for the dependency based method

\(^2\)Since we extracted only 84 facts for the location-of-birth relation using surface patterns in this case we evaluated 84 facts.
3.3. Experiments

with precision 93% means there must be at least $47261$ positive instances. So it is very likely that recall has improved.

We can do the same calculation for the equivalence rules method: $59830$ facts with precision 80% means there must be at least $47864$ positive instances. The interval overlaps with the interval for the dependency based method, so here we cannot be sure if recall improved even further.

Looking at the results for each relation separately, it is found that we can only credit three relations with the increase in number of facts extracted, namely function, founder and location-of-birth. The other three relations (capital, currency, and date-of-birth) show a decrease in the number of facts extracted. Especially the decrease for the date-of-birth relation is striking. The decrease for the currency table is negligible (only 1%) and although this cannot be said for the capital table (decrease of 7%), this relation shows at the same time a significant increase in precision seeming to indicate that the decline is mostly due to the fact that less noise is extracted using dependency patterns.

Finally, adding equivalence rules shows a positive effect overall. Especially for the founder relation, the date-of-birth relation, and the location-of-birth relation the increase was significant. They show a growth of 129%, 62%, and 55% respectively.

3.3.2 Question Answering task

Five separate question answering experiments were performed to investigate the differences in performance between different QA methods. We use the tables we have just presented in the previous section with extracted facts for the six question types capital, currency, date-of-birth, function, founder, and location-of-birth.

Questions are taken from the CLEF question sets as described in chapter 2. We had 18 capital questions, 10 currency questions, 8 date-of-birth questions, 15 founder questions, 96 function questions, and only 3 location-of-birth questions (See http://www.let.rug.nl/~mur/questionandanswers/chapter3/ for questions and answers).

As QA system we use Joost, the Dutch QA system which is based on information retrieval techniques. We incorporated the off-line module Qatar which uses for each experiment different sets of tables, which are described in the previous section. Joost and Qatar are described in section 2.1.1 at page 15 and following pages.

For those questions that are not answered by the off-line method (because no matching table entry was found), the QA system passes a set of keywords extracted from the question to the IR engine. IR returns a set of relevant paragraphs. Within
Chapter 3. Extraction based on dependency relations

<table>
<thead>
<tr>
<th></th>
<th>Joost</th>
<th>Surface</th>
<th>Dep. rels</th>
<th>Dep. rels + e</th>
<th>Dep. rels + e + d</th>
</tr>
</thead>
<tbody>
<tr>
<td>capital</td>
<td>0.483</td>
<td>0.861</td>
<td>0.944</td>
<td>0.944</td>
<td>0.958</td>
</tr>
<tr>
<td>currency</td>
<td>0.100</td>
<td>0.750</td>
<td>0.700</td>
<td>0.750</td>
<td>0.750</td>
</tr>
<tr>
<td>date of birth</td>
<td>0.125</td>
<td>0.750</td>
<td>0.375</td>
<td>0.750</td>
<td>0.750</td>
</tr>
<tr>
<td>founded</td>
<td>0.767</td>
<td>0.867</td>
<td>0.867</td>
<td>0.900</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>0.743</td>
<td>0.704</td>
<td>0.754</td>
<td>0.808</td>
<td>0.811</td>
</tr>
<tr>
<td>location of birth</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.167</td>
<td>0.111</td>
</tr>
<tr>
<td>total</td>
<td>0.619</td>
<td>0.726</td>
<td>0.744</td>
<td>0.805</td>
<td>0.811</td>
</tr>
</tbody>
</table>

Figure 3.3: Question answering results - number of questions answered correctly. + e refers to the addition of equivalence rules and + d refers to the addition of the d-score.

<table>
<thead>
<tr>
<th></th>
<th>Joost</th>
<th>Surface</th>
<th>Dep. rels</th>
<th>Dep. rels + e</th>
<th>Dep. rels + e + d</th>
</tr>
</thead>
<tbody>
<tr>
<td>location of birth</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.167</td>
<td>0.111</td>
</tr>
<tr>
<td>founded</td>
<td>0.767</td>
<td>0.867</td>
<td>0.867</td>
<td>0.900</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>0.743</td>
<td>0.704</td>
<td>0.754</td>
<td>0.808</td>
<td>0.811</td>
</tr>
<tr>
<td>currency</td>
<td>0.100</td>
<td>0.750</td>
<td>0.375</td>
<td>0.750</td>
<td>0.750</td>
</tr>
<tr>
<td>total</td>
<td>0.619</td>
<td>0.726</td>
<td>0.744</td>
<td>0.805</td>
<td>0.811</td>
</tr>
</tbody>
</table>

Figure 3.4: Question answering results - MRR. + e refers to the addition of equivalence rules and + d refers to the addition of the d-score.
this set, we try to identify the sentence which most likely contains the answer, and we try to extract the answer from the sentence.

The patterns defined for extraction online are not the same as the patterns defined for the off-line extraction. They are developed independently by different persons. The online answer extraction patterns can be more general since they are used to extract answers from passages detected in an earlier process step by the IR engine to be relevant passages rather than from the whole corpus and for this reason it is possible that Joost can still compute an answer online, when Qatar failed to find an answer.

For each sentence \( A \), we compute its \textbf{a-score} relative to the question \( Q \) as follows:

\[
\text{a-score}(Q, A) = \alpha \cdot d\text{-score}(Q, A) + \\
\beta \cdot \text{type-score}(Q, A) + \\
\gamma \cdot \text{IR}(Q, A) + \\
\delta \cdot \frac{1}{\text{FreqRank}}
\]

Here, \( d\text{-score} \) expresses to what extent the dependency relations of \( Q \) and \( A \) match as explained in section 3.2.2.2. In the experiments where we do not want to take this score into account it is set to 1 for all candidate answers. The \textbf{type-score} expresses whether a constituent matching the question type of \( Q \) could be found (in the right syntactic context) in \( A \), and \( \text{IR} \) combines the score assigned by the IR engine and a score which expresses to what extent the proper names, nouns, and adjectives in \( Q \) and \( A \) plus the sentence immediately preceding \( A \) overlap. The sentence preceding \( A \) is taken into account to approximate the technique of coreference resolution. The idea is that if terms from question \( Q \) appear in the sentence preceding \( A \), it is more likely that \( A \) contains the answer to \( Q \). \( \text{FreqRank} \) is the rank of a candidate answer when these answers are sorted according to frequency. \( \alpha, \beta, \gamma, \) and \( \delta \) are (manually set) weights for these scores.

This metric used for selecting the best answer from a list of results provided by IR can be used for re-ranking the results of table look-up as well. Since these answers are not found by IR, the score that would have been assigned by the IR engine is set to 1 for all candidate answers.

As a baseline we took the results we achieve by using only Joost to answer the questions. The off-line module Qatar is switched off. Results for this set of experiments are shown in the second column of table 3.3. Listed are the number of questions answered correctly. We have put the number of questions answered by Qatar between brackets. For the baseline where we use only Joost for question answering the number between brackets is of course always zero.
In the third column are the results achieved by using tables created by extracting facts with surface patterns. For the results in the fourth column we used dependency patterns. For the results in the fifth column we added the use of equivalence rules. Finally, the last column shows the results obtained when we take into account the \texttt{d-score} for answer ranking as well.

The first three experiments do not use equivalences over dependency relations. The first four experiments do not use \texttt{d-score} to re-rank answers. It should be noted, however, that the system still makes use of dependency relations for question analysis and answer extraction in these experiments.

Remarkably, we did not find a correct answer for a location-of-birth question in any of the experiments. However, there were only three questions for this relation, so we cannot draw any definite conclusions from this outcome.

Looking at the results for the complete set of questions (last row in table 3.3) we see a constant improvement. Using Qatar based on surface patterns 16 questions more are answered correctly. Replacing surface patterns by dependency patterns adds five questions. Adding equivalence rules improves the results with nine more questions in total (the number of questions answered by Qatar even went from 49 to 99). Finally, including the \texttt{d-score} for answer ranking resulted in two more questions being answered correctly, but here there was no effect for the questions answered by Qatar.

Looking at the results for each relation separately, we see that the use of Qatar based on surface patterns compared to using only Joost improves performance for the four relations capital, currency, date-of-birth, and founded. The result for location-of-birth stayed the same, and only for the function questions performance decreased (from 63 to 59 questions answered correctly).

On the other hand, using dependency relations instead of surface patterns did have a positive effect on the results for the function questions. Also the result for the capital questions improved, which is striking since the table with capital facts decreased in size, as we saw in the previous section. For the date-of-birth relation fewer questions were answered correctly which corresponds to the decrease of extracted facts for this relation. For the remaining relations the outcome did not change.

Equivalence rules turn out to be especially beneficial for answering function questions, but also the result for date-of-birth questions improved. For the founder relation more questions were answered with Qatar, but the total number of questions answered correctly did not change. Also for the remaining relations nothing changed.

Finally, the \texttt{d-score} helped a little as well. One more function question and one more founder question were answered correctly.
3.4 Discussion of results

To take into account the total number of questions per relation and also the top five answers the system returns per question instead of only the answer ranked first, we decided to calculate the mean reciprocal rank for each experiment. Calculating the mean reciprocal rank means that we took for each question the reciprocal of the rank at which the first correct answer occurred or 0 if none of the returned answers was correct. Then we took the average over all questions per experiment. The results are presented in table 3.4.

In general, the results of table 3.4 correspond to the results in table 3.3. However, in some cases small effects can be seen that remained hidden in table 3.3. For example, we now can see that using equivalence rules and applying the d-score both do have an effect on the location-of-birth questions.

3.4 Discussion of results

The results overall seem to indicate that the use of dependency patterns has a positive effect, both on the performance of the extraction task as well as on the question answering task. More facts are extracted compared to when surface patterns are applied and more questions are answered correctly.

We applied a set of equivalence rules to account for syntactic variation. This improved performance for off-line QA in particular. Since many more facts were extracted, the number of questions that can be answered by the off-line method increased a lot.

The results for adding the d-score as a factor in re-ranking answers were unfortunately less convincing. Although more questions were answered correctly they were not answered on Qatar’s account. This outcome is in accordance with the results in Bouma et al. (2005).

For the separate relations it turned out that performance did not always improve with the addition of more sophisticated techniques. Dependency relations for instance did not help for the capital, currency and date-of-birth relations. The reason why certain facts extracted by the surface pattern method were not extracted by the dependency pattern method can be explained for most part by parse errors.

The difficulty for capital questions can be illustrated by the following two examples:

(10)  a. Volgens de Turkse ambassadeur in de Belgische hoofdstad, Brussel [...]  
     English: According to the Turkish ambassador in the Belgian capital, Brussels [...]
Chapter 3. Extraction based on dependency relations

b. Volgens de Turkse ambassadeur in de Belgische hoofdstad, M. Keskin [...] English: According to the Turkish ambassador in the Belgian capital, M. Keskin [...]

Using a surface pattern as defined in appendix A for capital facts the correct capital fact ‘Belgisch-Brussel’ is extracted from sentence (10-a). In addition, the incorrect capital fact ‘Belgisch-M. Keskin’ is extracted from sentence (10-b). The ambiguity here is that in sentence (10-a) the appositive term (Brussel) depends on the head noun of de Belgische hoofdstad, and the appositive term in (10-b) (M. Keskin) depends on the head noun of de Turkse ambassadeur. According to Alpino both appositive terms, Brussel and M. Keskin, are in a dependency relation with the head noun of de Turkse ambassadeur. Since this did not match any of the predefined dependency patterns both facts were not extracted in the experiment where we used dependency patterns. It explains why we extracted more facts, but also more noise, for the surface pattern based method for the capital relation.

For the currency relation the difference in number of facts extracted was only small. Facts were missed because of simple parse errors. In sentence (11-a), for example, Duitse is not recognised as a adjective of marken. However, there are also examples where syntactic patterns turn out to be more robust to variation. The order in sentence (11-b) differs from the order we defined in the surface pattern, but it does not influence the extraction with dependency patterns.

(11) a. Als ermee in Duitsland wordt gebeld, wordt in Duitse marken afgeboekt. English: When one calls with it in Germany, German marks are being transferred.

b. De munt in Israël is de shekel [...] English: The currency in Israel is the new sheqel.

The text in which dates of birth were reported were pre-eminently suitable for surface patterns. Especially the pattern starting with a name followed by the birth year and year of death between brackets is best expressed in a simple regular expression. Syntactic patterns cannot compete with that.

For the three remaining relations, on the other hand, performance improved a lot when we replaced the surface patterns by the dependency patterns. A couple of examples for each relation show the wide variety in which facts can be expressed and which turned out to cause no problems for the dependency patterns, in contrast to the surface patterns method which extracted none of these facts.
First we give some sentences from which we extracted founder facts. We already explained the difficulties for the founder relation in the introduction, but here are some more examples from our experiments that support the idea expressed there.

(12) The founder relation.

a. Vaticaanstad, in 1929 opgericht [...]  
   English: Vatican City, founded in 1929 [...]  

b. [...] toen hij in 1964 de Televisie Radio Omroep Stichting oprichtte.  
   English: [...] when he founded the Television Radio Broadcast Foundation in 1964.  

c. [...] richtte hij in 1959 het Nederlands Dans Theater in Den Haag op.  
   English: [...] he founded the Dutch Dance Theater in 1959 in The Hague.  

d. De grondlegger van de Avro, Willem Vogt [...]  
   English: The founder of the Avro, Willem Vogt [...]  

e. Over oprichter Martin Batenburg van het Algemeen Ouderen Verbond [...]  
   English: About founder Martin Batenburg of the General Elderly Alliance

Function facts that were missed by the surface pattern method were often stated in the following order: function term, name of the person and then the country or the organisation. See (13-a) for an example. This order was not defined in a surface pattern. Yet the dependency patterns were flexible enough to match these kind of facts. Also problematic for the surface pattern extraction method was when there were too many terms between the function term and the person name, such as in (13-b) where we had to narrow the distance between minister and Johan Jorgen Holst and such as in (13-c) between Vladimir Zjirinovski and politicus.

(13) The function relation.

a. Premier Reynolds van Ierland [...]  
   English: Prime-minister Reynolds of Ireland [...]  

b. De Noorse minister van buitenlandse zaken Johan Jorgen Holst [...]  
   English: The Norwegian minister of foreign affairs Johan Jorgen Holst [...]  

c. Vladimir Zjirinovski, de extreem-rechtse Russische politicus, [...]  
   English: Vladimir Zjirinovski, the extreme-right Russian politician
For the location-of-birth facts it was also frequently the case that too many terms, most often the birth date, intervened between the location name and the person name ((14-a), (14-b), and (14-c)). Another tricky example is given in (14-d). To be born can be translated in Dutch with two different verbs: *geboren zijn* and *geboren worden*. We defined surface patterns using only one of these expressions, *geboren worden*. Therefore the fact in (14-d) which uses a form of *geboren zijn* was missed. For the dependency based method it did not matter since there is a dependency relation between *geboren* and the person name.

(14) The location-of-birth relation.

a. Theo Olof, in 1924 in Bonn geboren [...]  
   English: Theo Olof, born in 1924 in Bonn [...]  

b. Lucebert werd geboren op 15 september 1924 in Amsterdam.  
   English: Lucebert was born on the 15th of September 1924 in Amsterdam.  

c. Eise Eisinga, 250 jaar geleden geboren in Dronrijp, [...]  
   English: Eise Eisinga, born 250 years ago in Dronrijp, [...]  

d. Camus was geboren in Algerije.  
   English: Camus was born in Algeria

In general we can say that although facts are missed because of parse errors, it is still preferable to use dependency patterns instead of surface patterns for fact extraction. For a few relations it is more natural to use surface patterns as we saw for the date-of-birth relation. However, for many facts it holds that they are hard to deal with using surface patterns whereas they cause no problem for dependency patterns. In other cases more surface patterns are needed where only one dependency relation is sufficient.

Adding equivalence rules improved performance for every relation. Of course, for each relation, the number of extracted facts could have been increased by a similar amount by expanding the number of patterns for that relation. The interesting point here is that in this case this was achieved by adding a single, generic component.

The extracted facts were mainly due to two equivalence rules, namely the rule that accounts for relative clauses ((5-f) on page 43) and the rule that accounts for the switch in the order of the apposition relation ((5-a) on page 43). (15) list some examples of facts extracted using dependency patterns and additional equivalence rules that were missed by the method which used only dependency patterns.

(15) a. [...] de Anglicaanse Kerk, die door Hendrik VIII werd gesticht.
3.4. Discussion of results

English: the Church of England, which was established by Henry VIII.

b. [...] Carthago, de in de 9de eeuw vóór Christus door de Foeniciërs gestichte havenstad.

English: Carthage, the port founded by the Phoenicians in the 9th century before Christ.

c. President Mandela werd in 1918 in de Transkei geboren.

English: President Mandela was born in 1918 in the Transkei.

d. [...] de 17de-eeuwse schilder Aelbert Cuyp (1620-1691).

English: [...] the 17th-century painter Aelbert Cuyp (1620-1691).

e. Piet Mondriaan, die in 1872 in Amersfoort werd geboren, [...] 

English: Piet Mondriaan, who was born in 1872 in Amersfoort, [...] 

Also the problem with using dependency patterns compared to surface patterns now becomes more clear for the date-of-birth relation. The person name is often dependent on a function term, such as in examples (15-c) and (15-d), where the person names Mandela and Aelbert Cuyp are appositives of the function terms President and schilder respectively. That is why they were not matched by a dependency pattern. For a surface pattern it does not matter which terms are in front of the person name and therefore these facts are extracted by the surface pattern method.

The d-score improved performance only a little and not for the off-line question answering module. It is difficult to explain this result, but we suggest some possible reasons. First, there were only seven questions out of 39 questions that were not answered in the preceding experiment that contained modifiers of some sort, and consequently could benefit from adding the d-score.

Another point worth mentioning is that often the question did not contain any modifiers, but the answer sentence did, making it an incorrect answer. Example (16) shows this phenomenon. The question asks for the chairman of the soccer team Roma, implicitly it asks for the present chairman. The answer sentence, however, speaks about the former chairman. The noun voorzitter is modified by the adjective voormalige. In this case the candidate answer should receive a penalty for having extra modifiers. Further experimentation is needed here to investigate how we can make better use of this d-score.

(16) a. Question: Wie is de voorzitter van de voetbalploeg Roma?

   English: Who is the chairman of the soccer team Roma?

b. Answer: [...] de voormalige voorzitter van Roma, Gianmarco Calleri.

   English: the former chairman of Roma, Gianmarco Calleri.
3.5 Conclusion

The aim of this chapter was to compare extraction techniques based on surface patterns with extraction techniques based on dependency patterns, within the framework of off-line question answering. The hypothesis is that dependency patterns are more flexible than surface patterns, therefore more facts will be extracted and more questions are answered correctly. In short, recall will improve.

We defined two parallel sets of patterns, one based on surface structures, the other based on dependency relations, optimised both to the same degree. These sets of patterns were used to extract and collect facts, which we could use for off-line question answering.

The results of the experiments overall showed that the use of dependency patterns indeed has a positive effect, both on the performance of the extraction task as well as on the question answering task. Although there were some sentence structures that were most suitable for surface patterns, using dependency patterns increased both precision and recall in general. We can conclude that dependency relations eliminate many sources of variation that systems based on surface strings have to deal with.

Still, it is also true that the same semantic relation can sometimes be expressed by several dependency patterns. To account for this syntactic variation we implemented thirteen domain independent equivalence rules over dependency relations. Many more facts were extracted. For each relation, the number of extracted facts could have been increased by a similar amount by expanding the number of patterns for that relation. The interesting point is that in this case this was achieved by adding a single, generic component. Using these equivalence rules increased the number of questions answered by Qatar a lot (from 49 to 99 for 150 questions in total).

Finally, we introduced the d-score, which computes to what extent the dependency structure of question and answer match, so as to take into account crucial modifiers that otherwise would be ignored. Two more questions were answered by Joost, but further research is needed to fully exploit the information provided by this score.