Automatically classifying *why*-questions with the help of syntax[*]

Daphne Theijssen

This paper describes how syntactic information is used in a system that attempts to answer *why*-questions automatically. In this Question Answering system, machine learning algorithms classify *why*-questions according to their answer type (CAUSE or MOTIVATION), on the basis of deeper linguistic information such as syntactic structure. The aim of this paper is to establish how errors in the syntactic representations of the questions influence the performance of the *why*-question classifier. I employed the parsing system TOSCA (Oostdijk 1996) to obtain syntactic structures without human interference. From the classification accuracy we can conclude that the use of the parser module in TOSCA for distinguishing CAUSE and MOTIVATION is promising (86.8%) when provided with manually checked part-of-speech (word class) tags. Applying the whole TOSCA system – including POS tagging – fully automatically, however, seems troublesome due to the modularity of the system.

1. *Introduction*

For decades, linguists have tried to explain language use in terms of syntactic structure and have developed theories and models to describe language as accurately as possible. Also, languages have been compared in terms of their syntax. The goal of this paper is not to enrich our knowledge about syntactic structure, but to employ it in order to improve a language technology application.

Due to the rise of computers and especially the Internet, there is great demand for various language technological applications such as spelling checkers, search engines, machine translation systems, etc. One application within the field of information retrieval is that of Question Answering (QA). Where a regular search engine provides the user with a list of relevant documents, QA aims at answering a question in natural language by returning a list of possible answers taken from a set of texts, for example those on the Internet. Until recently the focus of

[*] In Elizabeth Koier, Olivia Loonen & Marieke Meelen (eds.), *Leiden Working Papers in Linguistics* 5.1 (2008), 36-52

http://www.lucl.nl ISSN 1574-4728

QA has been on closed-class questions, for example *who-*, *what-* and *where*-questions. The answer to these questions consists of a closed-class entity, for example a noun phrase. The type of answer that is expected is predicted by the question word used in the question: a *who*-question like *Who is the president of the U.S.A.?* expects a person as answer.

Much more complex methods are necessary when considering explanatory questions such as *why*-questions, which are currently addressed in the project 'In Search of the *WHY*' at Radboud University Nijmegen. An answer to a *why*-question can be a phrase, but can also be as long as a complete paragraph. If we want to find potential answers to a *why*-question, we need information of the type of answer to search for. The semantic answer type of *why*-questions can be summarised as REASON (Moldovan et al. 2000), but can be subdivided into CAUSE, MOTIVATION, CIRCUMSTANCE and GENERIC PURPOSE (Verberne et al. 2006). Since the *why*-QA system needs to function fully automatically, i.e., without human intervenience, it is necessary to classify the *why*-questions automatically according to their answer type.

Verberne et al. (2006) have developed such a module, employing machine learning algorithms. They show that knowing the syntactic structure of a question is beneficial to this task. Since their goal was to establish whether their syntax-based method would be useful for classification, they used manually constructed syntactic trees. In this paper I replicate Verberne et al.'s method with the difference that the syntactic trees are automatically determined. In order to establish the benefit of syntax to this fully automatic *why*-question classifier, the following questions need to be answered: How and how accurately can we automatically derive syntactic trees from raw text input? And how do inaccuracies in syntactic trees affect the performance of the classification?

In section 2, I describe the *why*-question classifier I based on Verberne et al. (2006). Section 3 describes how syntactic trees can be automatically obtained, and discusses the quality of the trees found. The trees were consulted for *why*-question classification, which is discussed in section 4. Finally, some concluding remarks and present suggestions for future research are presented in section 5.

2. Why-*question classification*

An important processing step in many Question Answering systems is the determination of the answer type. As mentioned in the introduction, the answer type is straightforward for some types of questions, e.g., a *who*-question expects a person (or other entity) as an answer. For other types, establishing the answer type is more complicated. *Why*-questions can be said to expect the answer type REASON. Verberne et al. (2006) argue this is best subdivided into CAUSE, MOTIVATION, CIRCUMSTANCE and GENERIC PURPOSE, because each subtype seems to be expressed with typical cues in the text (*due to* versus *in order to*, for example).

- The answer type is said to be CAUSE if there is a temporal causal relation between two events in which no deliberate human intention is involved. For example:
  Why did compilers of the OED have an easier time? – Because the OED was compiled in the 19th century when language was not developing as fast as it is today.

- The answer type is said to be MOTIVATION if there is a human intention involved in the temporal causal relation. A MOTIVATION can be either a future goal or some person's internal motivation. For example:
  Why has the team of researchers been split up into two teams? – To complete the work more quickly; one team will finish A while the second team will start on B.

- The answer type is said to be CIRCUMSTANCE if conditionality is added to the temporal relation: the first event is a strict condition for the second event. For example:
  Why will people buy Windows? – Because it offers more software, it is more fun to use and it works well enough.

- The answer type is said to be GENERIC PURPOSE if it does not express a temporal relation between two events, but gives the physical function of an object in the real world. For example:
  Why do people have eyebrows? – People have eyebrows to prevent sweat running into their eyes.

As in Verberne et al.'s (2006) approach, I will apply machine learning algorithms to classify the *why*-questions.

2.1. Data

I selected the data set for my experiments from a set of elicited *why*-questions provided by Verberne et al. (2006). To compile this question set, native speakers of English were asked to formulate *why*-questions to texts from Reuters' Textline Global News (1989), The Guardian on CD-ROM (1992) and Wall Street Journal (Penn Treebank) with the explicit mention that the answer to the question should be in the text. A semantic answer type was manually assigned to the questions, on the basis of classification guidelines. I randomly selected a subset of 13 source texts, leading to a set of 238 questions: 118 with answer type CAUSE, 117 with MOTIVATION and 3 with CIRCUMSTANCE. Since the subtypes CIRCUMSTANCE and GENERIC PURPOSE are infrequent (3 and 0 instances, respectively), in the following section I will focus on distinguishing between CAUSE and MOTIVATION. The number of questions (235) is quite small for machine learning experiments, but such small data sets are certainly not uncommon when there is a need for human involvement.

2.2. Relevant information

Intuitively, it seems that world knowledge is important for correct answer type classification. However, following Verberne et al. (2006), I assumed that lexical and syntactic information might also be sufficient. Partly on the basis of the original classification guidelines developed by Verberne et al. (2006), I defined 18 syntactic and semantic features.

The most salient feature seems to be the agentiveness of the subject (feature 1). If there is MOTIVATION, there must be somebody who causes the event described by the verb and this somebody is likely to be expressed as an animate subject, e.g., *Why did Bill Kerrey show up in Dixville Notch?* Similarly, the lexical verb can indicate how likely MOTIVATION is. Some verbs are anti-causative (2): they describe actions which are not caused by but simply happen to the subject, for example *to die* in *Why might thousands of women die if the present ruling is over-turned?* For both features, the information conveyed needs to be reinterpreted if the question is given with passive voice. Passive voice is therefore included as an additional feature (3). Another important feature is the modality of the verb phrase. Some modals, such as *should* or *mustn't*, seem to refer to MOTIVATIONS: *Why should we leave the law on abortion alone?* Others, such as *could* and *have to* are more likely to appear in questions asking for a CAUSE, as in *Why could FK-506 revolutionize the transplantation field?*. I included the presence or absence of *can* (4), *will* (5), *should* (6), *have to* (7), *might* (8), *shouldn't* (9), *could* (10), *would* (11), *shall* (12), *may* (13) and *ought* (14) as features to be used. I also suspected CAUSE to be signalled by specific syntactic constructions, namely intensive complementation (copular) (15), as in *Why is the research important?*, the lexical (non-auxiliar) verb *have* (16), for example *Why does Mr Tillotson have few peers when it comes to getting out the vote?*, and existential *there* constructions (17). Finally, I had an intuition that negation (18) on the finite verb might also be of use.

In order to establish the potential value of the features selected, I manually determined the values of the 18 features for all 235 questions in the data set (MANUAL). In other words, I checked each of these questions for the presence of an agentive subject, of an anti-causative verb, of passive voice, of modals, of lexical *have*, of existential *there* and of negation. Together with the correct answer type, they were offered to a machine learning algorithm. Such an algorithm tries to discover patterns in the data, which can again be used on unseen data so we can measure how the found model performs in predicting the answer type. When low accuracies are reached, it means that the features chosen are either insufficient, or the task is too difficult to perform automatically. High accuracies encourage the use of the features and the model for automatic classification.

The data from which the algorithm derives a model, is called the *training set*. The unseen data to which it is applied is referred to as the *test set*. Due to the rather small number of *why*-questions in the data set, I decided to apply 13-fold cross-validation on all cases, each time testing on questions for a specific source text and training on all other questions. I used the TiMBL system (Daelemans et al. 2004), to be exact version 5, using its default settings, i.e., an IBk learner with k=1.

To determine the asset of the aforementioned features for the classification in CAUSE and MOTIVATION, I determined what results can be obtained without any deeper linguistic information. For this *baseline*, I used a feature set consisting simply of all (776) words in the questions (signalling the presence or absence of the word).

2.3. Results

Answer type determination with the full sets of features yielded the classification scores in the second column (*all features*) in Table 1.

Table 1. Accuracy reached by TiMBL

feature set	all features	selection (all texts)	selection (training sets)
words (baseline)	51.5%	81.3%	80.4%
linguistic features	83.0%	83.8%	80.4%
words + linguistic features	74.0%	88.9%	78.3%

It seems that linguistic analysis can indeed be a great asset in determining the answer type for these questions since the accuracy increases when deeper linguistic information is added. The accuracy of the baseline is only 51.5%, while it is 83.0% when using the linguistic features. One would expect that TiMBL would classify even better when provided with both the baseline features (the words) and the linguistic features. This, however, is not the case (74.0%).

Since the linguistic features alone yielded better performance than the set of all features, I have to conclude that either TiMBL was overwhelmed by the large number of features, or that it considered irrelevant features important because they appeared to be so in the (too) small data set (overtraining). In order to arrive at more interpretable measurements, I implemented a simple feature selection mechanism that reduces the number of features, which I then applied in two different ways[1]. First, selection was done using the complete question set. As this used the same training-test split as the final quality test, the selected features were optimally geared for that final test and the final results (shown in the third column of Table 1) should be interpreted as a kind of upper bound. In the second application, the training set for a specific test (in the 13-fold cross-validation) was itself subjected to a 12-fold split to determine the optimal feature selection. Here, the test set for the final measurement was not used and the experiment was closer to normal methodology. However, given the small size of our data set, I assume that important features have been missed in the selection and that the results (shown in the fourth column of Table 1) should be seen as a lower bound.

The figures in Table 1 show that the baseline is much better (81.3% for the upper bound and 80.4% for the lower bound, compared to 51.5% without selection). From these differences, we can conclude that TiMBL was indeed affected by the abundance of features. For the upper bound, including both words and the linguistic features led to better performance (88.9%) than using them separately. Linguistic information proved better (83.8%) than lexical information (the words: 81.3%). For the lower bound, overtraining prevented optimal selection, so that the selection from the linguistic features led to worse results (80.4%) than the full set in the second column (83.0%). The selection from all possible features performed even worse (78.3%) than the selection from smaller sets (80.4% for both). Due to the overtraining problems, a clear assessment of the added value of linguistic analysis is not possible, but it seems promising.

With the proper selection method (the lower bound), I have established a new baseline: the 80.4% for the selected words. In the next section, it will be used to judge whether linguistic features are still beneficial when derived automatically.

[1] Van Halteren, personal communication

3. *Extracting syntax automatically*

In this section[2], I will try to answer the question how and how accurately syntactic trees can be derived from raw text input. Systems developed for the automatic extraction of syntactic structure are called syntactic *parsers*. The parser examined in the present study is the TOSCA system (Oostdijk 1996), an interactive syntactic parser that yields very detailed analyses of English text. The parser module is rule-based and the formal grammar underlying it is based on the descriptive system proposed by Aarts & Aarts (1982), which is an adaptation of the English grammar by Quirk et al. (1972).

An example of a syntactic analysis by TOSCA is presented in Figure 1. TOSCA analyses are constituency trees and essentially include three types of information: (1) information pertaining to the categorial realization of constituents (article, noun phrase, clause, etc.), (2) information about the functional role of a constituent (prepositional complement, subject, etc.) and (3) additional information (for example about the word order observed or the subclass of a particular word class) which is presented in the form of attributes. The three levels of analysis interfere with each other: incorrect categories and/or attributes may lead to the erroneous assignment of function labels to constituents, for instance because the distribution of thematic roles (e.g., direct object) depends on verb transitivity.

Figure 1. Example of TOSCA output for the question *Why are the attorneys for the Bush administration present at the hearing?*

```
NOFU,TXTU ()
├── UTT,S (act,indic,intens,inter,pres,unm)
│    ├── A,AVP (inter)
│    │    └── AVHD,ADV (inter) – Why
│    ├── INTOP,LV (indic,intens,pres) – are
│    ├── SU,NP ()
│    │    ├── DT,DTP ()
│    │    │    └── DTCE,ART (def) – the
│    │    ├── NPHD,N (com,plu) – attorneys
│    │    └── NPPO,PP ()
│    │         ├── P,PREP () – for
│    │         └── PC,NP ()
│    │              ├── DT,DTP ()
│    │              │    └── DTCE,ART – (def) – the
│    │              └── NPHD,N (com,sing) – Bush administration
│    └── CS,AJP (prd)
│         ├── AJHD,ADJ (prd) – present
│         └── AJPO,PP ()
│              ├── P,PREP () – at
│              └── PC,NP ()
│                   ├── DT,DTP ()
│                   │    └── DTCE,ART (def) – the
│                   └── NPHD,N (com,sing) – hearing
└── PUNC,PM (inter,qm) – ?
```

The TOSCA system consists of two automatic components, being a part-of-speech (POS) tagger

[2] This section is a comprised version of Theijssen et al. (2007)

and a parser. From now on, the terms ‘tagger’ and ‘parser’ are only used to indicate these particular automatic components, while the total system in which both are embedded is referred to as ‘TOSCA’ or the ‘(TOSCA) system’. Input to the system are text inputs that usually take the form of sentences. These are tagged automatically with POS (word class) information. The POS tagger is probabilistic and has been trained on a manually annotated corpus. The probabilities are based on the frequency of observed word classes and the immediate context (trigrams) of each individual token in the corpus. The tag set is elaborate: it includes the basic word classes such as ‘article’, ‘preposition’, ‘noun’, etc., but also further subclassifications for most word classes. ‘Verbs’, for example, can be subdivided according to their complementation type (transitivity) and form (tense, mode and number) (Van Halteren & Oostdijk 1993). The human analyst working with the system verifies whether with each of the tokens in the input string the correct tag is associated. Moreover, where required, the analyst inserts syntactic markers that help reduce the degree of ambiguity of highly ambiguous strings such as prepositional phrases and coordinated constituents. The unambiguously tagged input along with the syntactic markers that have been added is then submitted to the parser. Erroneously selected POS tags greatly influence the range of possible syntactic structures that can be yielded by the parser. The monotransitive form of *decline* in *Why did the Cincinnati Public schools decline to carry the program?*, for example, might be incorrectly tagged as an intransitive verb. Consequently, the clause *to carry the program* cannot be classified in any of the available syntactic structures because the verb attribute ‘intransitive’ prevents the assignment of the correct function to this direct object.

Since the parser has no knowledge of the contextual (semantic, pragmatic and extra-linguistic) knowledge that is called upon, it generates all possible syntactic analyses. However, it includes a penalty system that favours certain – for humans – intuitively more appropriate analyses than others. It prefers, for example, unmarked word order over marked word order. Still a number of parses with equal penalties may remain, from which the human analyst is expected to select the one correct analysis for storage in a linguistic database. For more details on the TOSCA system, the reader is referred to Van Halteren & Oostdijk (1993).

The structure of TOSCA enables investigating how reducing the human intervention influences the quality of the parser output. In this way, the value of the separate TOSCA modules can be estimated for the parse quality and, eventually, for the classification of *why*-questions.

3.1. Data

In the previous section, the TOSCA system has been described as an interactive system, consisting of four different stages: (1) automatic tagging, (2) manual tag correction and syntactic marker insertion, (3) automatic parsing and (4) manual parse selection. I derived three data sets from the 238 questions described in Section 2.1.:

- a gold standard (from now on referred to as GOLD)
- a semi-automatic output (SEMI), in which I applied tag correction and manual insertion of syntactic markers

- a fully automatic output (AUTO), in which only the two automatic components (POS tagger and parser) are used.

Going through all four steps in the interactive TOSCA system, GOLD was developed. This means that the POS tags were manually corrected, syntactic markers were manually inserted and the best analysis was manually selected from the proposed parses. For questions that could not be parsed despite our intervention after the tagging and parsing stages, I manually created gold standard trees. SEMI has been obtained by employing the interactive TOSCA system as it was meant up until the actual parsing process. Often, the parser proposed more than one possible syntactic analysis. The order in which these parses are presented is not based on linguistic theory but depends on the system's procedure of passing through the grammatical rules. For SEMI, I always saved the first proposed tree, which is neither ranked first nor completely randomly selected by the parser. To create AUTO, the list of tags proposed by the POS tagger and the first tree proposed by the parser were left unchanged. In this set-up no syntactic markers are inserted because this would involve changing the system (the insertion of syntactic markers presently requires manual intervention on the part of the human analyst; the alternative of producing a script that guesses the location of the markers would be possible, but would alter the system).

3.2. Parser evaluation

In order to measure the robustness of the parser, I calculated the proportion of questions for which the parser was able to produce output (the coverage) for both SEMI and AUTO.

To establish the quality of the parser output, I used Parseval, which is the common metric for evaluation of the quality of constituency trees. Parseval is also referred to as GEIG (Black et al. 1991). Parseval's evaluation method is based on lining up the brackets delimiting constituents. A sentence *a b c* with a gold standard *[a b] c* for instance, is considered not structurally consistent with an output *a [b c]*, because there is a crossing error (Black 1993). In addition to the average number of crossing brackets, precision and recall are calculated. The precision is a ratio of the number of correct brackets in the system's parse to the total number of brackets in the system's parse, while the recall is a ratio of the number of correct brackets in the system's parse to the total number of brackets in the gold standard. Following Van Rijsbergen (1979), the F-score can be calculated, which represents the harmonic mean of precision and recall. Since Black (1993), the Parseval metric has been extended. Magerman (1995) has decided to include the assignment of labels in the metric. For example, if the gold standard is *[PP [P a] [NP b]] [VP c]* and the parser output *[ADV a] [VP [V b] [V c]]*, the evaluation is based on comparisons of the location of the brackets as well as the choice of labels. This has led to the measures 'labelled precision' and 'labelled recall'.

The Parseval metric is useful to measure the quality of the syntactic parse, but is not helpful in pinpointing where exactly errors are made. Therefore, I also employed the approach proposed by Sampson et al. (1989), and further discussed by Sampson (2000), which is Leaf-Ancestor Assessment (LA). The calculation of the LA score can best be explained by means of an example. Figure 2 shows a syntactic tree of the question *Why are 4300 additional teachers required?*, in which *4300 additional* and *teachers* have been incorrectly analysed as two separate NP's.

Figure 2. Example of a syntactic tree: *Why are 4300 additional teachers required?*

```
NOFU,TXTU ()
├── UTT,S (act,indic,intens,inter,pres,unm)
│   ├── A,AVP (inter)
│   │   └── AVHD,ADV (inter) – Why
│   ├── INTOP,LV (indic,intens,pres) – are
│   ├── SU,NP ()
│   │   └── NPHD,NUM (frac,sing) – 4300 additional
│   ├── CS,NP ()
│   │   └── NPHD,N (com,plu) – teachers
│   └── A,CL (-su,intr,pass,pastp,unm,zsub)
│       └── V,VP (indic,intr,pass,pastp)
│           └── MVB,LV (indic,motr,pastp) – required
└── PUNC,PM (inter,qm) – ?
```

Starting from a terminal element, being a leaf in the tree, one moves up in the tree and registers each node label of the desired information level until one reaches the root of the tree. If necessary, squared brackets are inserted in the label sequence to delimite branches with multiple nodes. For *4300 additional*, for example, the category label sequence is *NUM NP S TXTU* (numeral, noun phrase, sentence, text unit). Similarly, a category label sequence can be determined for *4300 additional* in the correct syntactic analysis, which should include brackets because *4300 additional teachers* is a multi-node branch: *NUM [NP S TXTU*. The one label sequence is then changed into the other by deleting, inserting and substituting labels. Deletion and insertion is penalised with a penalty of 1, substitution with a penalty of 2 (since it consists of a deletion and an insertion). From the range of possible ways to arrive at the desired label sequence, the procedure with the *minimum edit distance* (the lowest penalty total) is selected. The minimum edit distance for the two label sequences mentioned is 1 (being a deletion of the bracket). The LA score is calculated by subtracting the minimum edit distance from the total number of labels (including brackets) in output and gold standard together, and dividing this again by the total number of labels and brackets. In the example the LA score is (9-1)/9 = 0.89. Combining the scores for all terminal elements indicates the score for the whole sentence. Likewise a score can be determined for the whole data set.

Theijssen et al. (2007) shows that the Parseval F-score and the Leaf-Ancestor Assessment score are highly correlated for the data introduced in Section 3.1. The similarity provides enough support to use either method, depending on which suits the evaluation purpose best. I will apply coverage and the Parseval metric to both AUTO and SEMI to discover the effect of human involvement and to enable other researchers to compare the results to other syntactic parsers. Sampson's LA measure is useful when one wants to discover which clauses or constructions are problematic for TOSCA. In the beginning of this section, I have remarked that tagging errors have severe consequences for the parser performance. Since the focus of this paper is on syntax, not on POS tagging, it would be undesirable to do an error analysis on parser output that is based on false information (the incorrect POS tags). For this reason, I do an error analysis on SEMI only, using the LA measure.

3.3. Results

The coverage, the number of perfect matches and the Parseval scores are presented in Table 2. From the set of 238 questions, TOSCA was able to parse 233 in SEMI, and only 190 in AUTO. Of 233 questions in SEMI, 188 were a perfect match with GOLD, compared to only 41 of 190 trees in AUTO. AUTO achieves a lower precision and recall and has more crossing brackets than SEMI (the differences in Parseval scores are significant ($p < 0.001$) following the independent t-test). In AUTO, 84.5% of the POS tags including their specifications (*V(intr, inf)* for example) is completely correct for this data set.

Table 2. Tag accuracy, coverage, perfect match and Parseval scores for SEMI and AUTO

	SEMI	AUTO
Tag accuracy	100%	84.5%
Coverage	99.1% (230 of 235)	80.9% (190 of 235)
Perfect match	80.7% (188 of 233)	21.6% (41 of 190)
Labelled Precision	96.0%	79.4%
Labelled Recall	95.7%	77.2%
Labelled F-value	95.9%	78.3%
Average nr crossing brackets	0.060	0.310

The differences between SEMI and AUTO in the table confirm the expectation that the accuracy of the tags provided to the parser is essential for the performance of the TOSCA system. This is obvious since the parser is designed so as to produce (minimally) the correct parse on the basis of correctly tagged input. Erroneously tagged input will cause the parser to fail to produce a correct parse. Thus, human intervention is required to manually correct any erroneous tags resulting from the application of the POS tagger.

In more than 80% of the covered questions in SEMI, there is no need for the human analyser to select the correct syntactic tree, since it is presented first (80.7% perfect match). Taking into account the fact that the parser does not include a ranking procedure for trees that have obtained equal penalties during the parsing process, I consider this percentage of perfect matches rather large. It encourages a fully automatic use of the parser (the second automatic component of the TOSCA system) for the purpose of *why*-question classification.

The LA scores for the TOSCA parses in SEMI are presented in Table 3. The differences between the scores for categories, functions and attributes are significant ($p < 0.001$ for all three pairs, following the paired (dependent) t-test). The scores for categories are highest, those for functions lowest. As established in the previous section, more than 80% (188 questions) of the parses are a complete match of the gold standard.

Table 3. LA scores for TOSCA output in SEMI

	Categories	Attributes	Functions	Average
LA Score	98.8%	98.3%	97.6%	98.2%

3.4. Error analysis

Despite the fact that the parser is a wide-coverage parser intended to parse unrestricted input, I found that for 5 questions in our data set it was unable to produce an analysis, even when provided with gold standard tags (SEMI). Two questions included a coordination that apparently was too complex, another two were problematic because of the percent symbol (%) and one question included a date (*April 26, 1990*). For AUTO, the same problems occurred, except for the last-mentioned, where a tree could be produced due to tagging errors. However, in AUTO another 44 questions could not be parsed. For a detailed error analysis, I refer to Theijssen et al. (2007).

Now let us turn to Sampson's Leaf-Ancestor Assessment. In 36 of the 233 questions that could be parsed, there was an error in the placing of brackets. Brackets are only placed when a node has one or more sisters, so an incorrect placement of brackets is a straightforward clue for an erroneous tree structure. Another sign of an imperfect tree structure is the lack or surplus of node labels in a sequence. This was the case in the same 36 questions plus one other.

An example of an incorrect analysis is that yielded for the question *Why are films planned for release only overseas?*, in which *planned ... overseas* is incorrectly parsed as a postmodifier of the noun *films* (Figure 3). Ignoring modifiers, the question in the parse is *Why are films?*. The *films* are further specified as those that are *planned for release only overseas*. The Leaf-Ancestor Assessment score can be employed to trace these errors. The word *planned* in Figure 3, for instance, has the (category) sequence *LV VP [CL NP S TXTU* (lexical verb, verb phrase, [, clause, noun phrase, sentence, text unit). Comparing this to the gold standard (*LV VP S TXTU*), we see that the use of brackets fails and there is a lack of nodes. Both observations help in establishing that the tree structure is erroneous and in locating in what part of the tree the errors occurred.

Figure 3. Example of TOSCA output in SEMI for the question *Why are films planned for release only overseas?*

```
NOFU,TXTU ()
├── UTT,S (indic,inter,intr,pass,pres,unm)
│    ├── A,AVP (inter)
│    │    └── AVHD,ADV (inter) – Why
│    ├── INTOP,AUX (indic,pass,pres) – are
│    └── SU,NP ()
│         ├── NPHD,N (com,plu) – films
│         └── NPPO,CL (-su,indic,intr,pass,pastp,unm,zero)
│              ├── V,VP (indic,intr,pass,pastp)
│              │    └── MVB,LV (indic,motr,pastp) – planned
│              ├── A,PP ()
│              │    ├── P,PREP () – for
│              │    └── PC,NP ()
│              │         └── NPHD,N (com,sing) – release
│              └── A,AVP (gen)
│                   ├── AVPR,AVP (excl)
│                   │    └── AVHD,ADV (excl) – only
│                   └── AVHD,ADV (gen) – overseas
└── PUNC,PM (inter,qm) – ?
```

Substitutions of node labels demonstrate incorrect label selection. They especially occur for the attributes and functions selected by the TOSCA parser and to a less extent for categories. In 7 questions, the clause tense was incorrect, for instance by mistaking a progressive construction for a present participle construction. In a few other questions (3), there were problems concerning modality or voice. Errors in the functions 'subject', 'subject complement', 'direct object' and 'adverbial' occur in 28 questions. Of these 28 questions, the transitivity of the main clause was wrong in 5 questions, in all of which a monotransitive main clause was erroneously parsed as an intransitive one. Since the parser was offered manually checked tags, the transitivity of the verb in the parse must be correct. The problem is that the monotransitive verb is erroneously placed in a subclause, making the subclause monotransitive and the main clause intransitive. This again leads to an erroneous assignment of the function labels 'subject' and 'adverbial' to elements in the non-existent subclause. In 9 questions, the question word *why* was incorrectly parsed as a subject complement instead of an adverbial. Because of this, the word order feature 'pre-cs' instead of 'unmarked' is selected, meaning the fronting of a subject complement. The remaining 14 questions involving the functions mentioned have too diverse causes to describe here.

Table 4 shows a list of words obtaining an LA score lower than 1 (being the perfect score). The first number shows the proportion of occurrences with an erroneous label sequence and the second the average LA scores obtained for all occurrences of the word. The LA scores are the average of the scores obtained for category, function, and attribute(s). I have only listed words that have a frequency of at least five, of which at least a quarter has an imperfect label sequence. This decision prevents inclusion of unique or rare words that have an imperfect analysis: if a word occurs only once in the data set and its label sequence contains an error, 100% of this word fails, which would undesirably position it high in the list.

Table 4. Words with an imperfect label sequence

word	prop.	LA	word	prop.	LA	word	prop.	LA
than	60%	80%	*dictionary*	40%	89%	*at*	30%	88%
chefs	60%	82%	*with*	38%	76%	*women*	29%	70%
for	47%	77%	*about*	33%	83%	*and*	29%	88%
court	44%	80%	*warming*	33%	88%	*up*	25%	81%
supreme	43%	79%	*rights*	33%	88%	*in*	25%	84%
easier	40%	68%	*global*	33%	91%			

The word list enables us to locate difficulties in parsing the data I used. Interesting is the large number of prepositions in this list despite the fact that for the greater part, PP-attachment is determined by syntactic markers. These have been manually inserted prior to the parsing process. The list also shows word groups that occur in the same questions. The words *dictionary*, *easier* and *than*, for instance, are all used in questions posed to a newspaper text about compiling a Spanish equivalent of the Oxford English Dictionary (OED). It appears that though formulated by different native speakers of English, the questions have a similar structure. This is likely to be caused by the design of the elicitation experiment, where participants had access to the news paper texts while formulating questions to them. In questions to other texts, co-occurring words are *court*, *supreme*, *rights* and *women*, and *warming* and *global*. It was beyond the scope of the present evaluation to include employing a larger data set with more syntactic and lexical diversity to verify whether the results at the word level are representative for *why*-questions in general.

4. *Automatic classification*

In the previous section we have seen how the TOSCA parser can be employed to automatically derive syntax, and how accurate the parser output is. When applied fully automatically, the Parseval labelled F-scores for the questions that could be parsed are much lower (78.3%) than those reached when the tags are corrected and the necessary markers are inserted (95.9%). With the help of the Leaf-Ancestor Assessment score, I have also been able to determine what errors are made by the TOSCA parser (see Section 3.4).

As opposed to the procedure presented in Section 2, where the features were manually determined and can therefore be assumed to be flawless, they will have to be derived fully automatically to make the answer type classifier useful for *why*-Question Answering. The data sets GOLD, SEMI and AUTO are employed for the automatic feature extraction. Again, only the questions with the answer type CAUSE or MOTIVATION are included (235 questions). The found results will help in establishing how the parse errors influence the performance of the *why*-question classifier.

4.1. *Feature extraction*

For the derivation of all the linguistic features previously described, deep linguistic processing was needed. The values of the features for the questions were extracted by consulting WordNet (Fellbaum 1998) for subject agency, Verbnet (Kipper et al. 2000) for declaratives, the Levin verb index (Levin 1993) for anti-causatives and the syntactic structures from TOSCA (Oostdijk 1996) for voice, intensive complementation, declaratives and existential *there*. Negation was determined by checking whether the question contained *not* or *n't* and modality by looking up whether the auxiliary of the question was in the list of modals. When a question had *have*, *having* or *had* as its main verb, it was considered a lexical *have* question. All feature values were gathered by using a Perl-script.

Problematic was the ambiguity in declarative layer questions, for example: *Why does Mr. Bocuse say he will use the damages to build a cooking school?* (Verberne et al. 2006:5). The question could either be why Mr. Bocuse said it, or why he will use the damages to build a cooking school. Declarative layer questions are recognised by checking whether the main verb is declarative (Verbnet) and the direct object is a subclause (TOSCA). The decision on the topic of the question is made on the basis of the semantics of the declarative verb, which is also taken from Verbnet. When this verb is factive, presupposing the truth of its complements, e.g., *know*, the Perl-script takes the other feature values from the main clause (*Why does Mr. Bocuse say...*). In declarative layer questions with a non-factive declarative verb, e.g., *think*, the feature values are drawn from the subordinate clause (*Why will he use the damages to build a cooking school?*).

For the questions that were not covered in SEMI and AUTO, Verberne et al. (2006) developed additional procedures that avoided the need for syntax and instead depended on the POS tags (either manually checked, as in SEMI, or not, as in AUTO). They wrote a new Perl-script that recognised patterns in the tagged questions, and used them to locate the subject, auxiliary and main verb. The features for which TOSCA was not used in the earlier procedure – negation, modality, verb type, lexical *have* and existential *there* – were treated as described above. The other feature values were found in a rather crude way. When a question had a form of *to be* as its auxiliary and the main verb ended in *-ed*, the voice was considered passive, in all other questions active. When the main verb was a form of *to be*, or the auxiliary was a form of *to be* and there was no main verb, the value of the feature intensive complementation was present. For declarative layer questions, TOSCA could not be consulted for determining whether the direct object is a subclause or not. Therefore, all questions with a declarative verb (according to Verbnet) were considered declarative layer questions. Although the feature values were more difficult to ascertain using the POS tags instead of the parser, Verberne et al. (2006) believe the resulting data set is optimal considering the information sources used.

The features that need syntactic information have been determined on the basis of the data sets GOLD, SEMI and AUTO, supplemented with the outcome of the robustness module if necessary. Each of the sets has been offered to TiMBL as in Section 2, except that only the first selection method is applied, in which the optimal feature selection is based on the whole data set. The reason for this choice is to prevent overtraining due to the smallness of the data set.

4.2. Results

The accuracies reached are presented in Table 5. The results indicate that errors made in the feature extraction decrease the accuracy reached when applying linguistic features (from 83.0% to 77.4%). This means that even with correct syntactic information, the feature extraction script is not able to derive the feature values perfectly. Selecting the most useful features from the words present and the linguistic features apparently filters the erroneous features: the difference is much smaller (88.9% and 86.8%, respectively). In this manner, the results surpass the baseline, which makes it promising to use linguistic information by the feature extraction method described, provided that the words are also included and an optimal selection (filtering) is made.

Table 5. Accuracy reached by TiMBL

feature set	linguistic features	selection words + linguistic features
baseline	80.4%	80.4%
MANUAL	83.0%	88.9%
GOLD	77.4%	86.8%
SEMI	77.9%	86.8%
AUTO	66.4%	82.1%

For the gold and near-gold parses, the quality was practically the same, and better than the quality of the word-based baseline (for the optimal selection). For the fully automatic parses, there was only a slight improvement (1.7%) over the baseline.

Now let us turn to the second question I posed in the introduction: How do inaccuracies in syntactic trees affect the performance of the classification? Table 6 shows the tag accuracies and Parseval F-scores for the three data sets GOLD, SEMI and AUTO, together with the accuracy reached by TiMBL when offered with the optimal set of features (both linguistic and word-based) derived from their syntactic trees.

Table 6. Summary of results

feature set	Tag accuracy	Parseval F-score	accuracy TiMBL (optimal feature set)
baseline	-	-	80.4%
MANUAL	-	-	88.9%
GOLD	100%	100%	86.8%
SEMI	100%	95.9%	86.8%
AUTO	84.5%	78.3%	82.1%

We see that when the TOSCA parser is provided with the correct tags and syntactic markers, the errors in the parser output have no influence on the accuracy reached by the *why*-question classifier when provided with the optimal selection of words and linguistic features. All errors and difficulties found with the Leaf-Ancestor Assessment metric in Section 3.4 are thus irrelevant in the given task. In other words: the performance of the TOSCA parser without manual verification is sufficient to be applied automatically in the *why*-question classifier.

When employing the whole TOSCA system (including the tagger) without any human involvement, less convincing results were obtained. The coverage was only 80.9%, and thus for almost 20% of the questions the feature values had to be extracted with the help of the robustness module described in Section 4.1. As we saw in Section 3.4, the incorrect tags and the missing syntactic markers lead to a high percentage of incorrect parse trees (almost 80%) and a relatively low Parseval F-score (78.3%). It is therefore not surprising that the parses are too erroneous (if even available) to still be convincingly useful for the given task.

5. *Conclusion*

In this paper, I have presented a method to classify *why*-questions according to their answer type with the help of automatically derived syntactic information. The answer type of *why*-questions can be summarised as REASON but is best subdivided, for example in CAUSE and MOTIVATION as is performed here. Previous research by Verberne et al. (2006) has already shown that deeper linguistic information such as syntactic structure is beneficial for the answer type classification task. The research presented followed their approach and complements it in two ways.

First, I constructed a set of manually determined features in order to check the quality of the feature extraction script. The results indicate that errors made in the feature extraction decrease the accuracy reached when applying linguistic features (from 83.0% to 77.4%). Selecting the most useful features from the words present and the linguistic features apparently filters the erroneous features: the difference is much smaller (88.9% and 86.8%, respectively). Since feature extraction is a crucial step in the *why*-question classifier, it is certainly advisable to attempt to improve the script in the future.

Second, I replaced the manually established syntactic trees used in their research by the trees produced by the syntactic parser TOSCA (Oostdijk 1996). TOSCA is an interactive parsing system that aims to yield deep linguistic analyses. The output includes detailed syntactic information in the form of categories, functions and attributes. The level of detail and the interdependence between the different types of information in the descriptive model that is being used entails the risk of causing a domino effect in which incorrect categories and/or attributes lead to the erroneous assignment of function labels to constituents. When provided with correct POS tags and post-edited input, however, more than 80% of the first proposed TOSCA analysis is a perfect match of the gold standard. The parses obtain an average Parseval score of 95.9%. When providing the *why*-question classifier (the feature extraction scripts and TiMBL) with an optimal selection of the words found and the linguistic features, the errors made by TOSCA in semi-automatic mode have no effect on the accuracy reached (86.8%, same as for the gold standard). This is a clear improvement over the word-based baseline accuracy of 80.4% and therefore encourages the use of the TOSCA parser in the question analysis component of the *why*-QA system.

The modularity of the current TOSCA system is fatal: tagging errors and missing syntactic markers in automatically obtained input radically decrease the coverage, showing that the parser is not at all robust. Moreover, the Parseval labelled F-scores for those questions that could be parsed were much lower (78.3%) than those reached when the tags are corrected and the necessary markers are inserted (95.9%). Due to the errors and the lack of available trees for almost 20% of the questions, TiMBL could only classify 82.1% of the questions correctly. This is only slightly better than the baseline, and gives little support for applying the TOSCA system (including tagger) in automatic *why*-question classification. A new version of TOSCA is under construction, in which the level of detail in the parses is maintained, while there is no longer a need to separately provide POS tags for the tokens in the input or insert any syntactic markers.

On the basis of the presented results, we can conclude that the use of automatically derived syntactic structures for the purpose of *why*-question classification is promising. The modularity in TOSCA is problematic, but will be solved in the near future.

Acknowledgements

I would like to thank Drs. Suzan Verberne for providing me with her data and for writing the feature extraction scripts. Furthermore, I thank Dr. Nelleke Oostdijk for her support with the TOSCA system, and Dr. Hans Van Halteren for his assistance in the machine learning experiments.

Daphne Theijssen
Department of Linguistics
Radboud University Nijmegen
d.theijssen@let.ru.nl
Project website: *http://lands.let.ru.nl/˜sverbern/*

References

Aarts, F. & J. Aarts (1982). *English Syntactic Structures*. Pergamon (Oxford).

Black, E. (1993). Statistically-based computer analysis of English. Black, G. R., E. & G. Leech (eds.), *Statistically-driven computer grammars of English: The IBM / Lancaster approach*, pp. 1–16.

Black, E., S. Abney, S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini & T. Strzalkowski (1991). Procedure for quantitatively comparing the syntactic coverage of English grammars. *Proceedings of the workshop on Speech and Natural Language*, Leiden, pp. 306–311.

Daelemans, W., J. Zavrel, K. Van Der Sloot & A. Van Den Bosch (2004). TiMBL: Tilburg Memory Based Learner, version 5.1. Tech. Rep. 04–02, ILK Research Group, Tilburg, The Netherlands.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Kipper, K., H. Trang Dang & M. Palmer (2000). Class-based construction of a verb lexicon. *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX.

Levin, B. (1993). *English Verb Classes and Alternations – A Preliminary Investigation*. The University of Chicago Press.

Magerman, D. (1995). Statistical decision-tree models for parsing. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Morgan Kaufmann, Cambridge, pp. 276–283.

Moldovan, D., S. Harabagiu, R. Pasa, R. Mihalcea, R. Grju, R. Goodrum & V. Rus (2000). The structure and performance of an open domain question answering system. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pp. 563–570.

Oostdijk, N. (1996). Using the TOSCA analysis system to analyse a software manual corpus. Sutcliffe, R., H. Koch & A. McElligott (eds.), *Industrial Parsing of Software Manuals*, Rodopi Amsterdam, pp. 179–206.

Quirk, R., S. Greenbaum, G. Leech & J. Svartvik (1972). *A Grammar of Contemporary English*. Longman (London).

Sampson, G. (2000). A proposal for improving the measurement of parse accuracy. *International Journal of Corpus Linguistics* pp. 53–68.

Sampson, G., R. Haigh & E. Atwell (1989). Natural language analysis by stochastic optimization: a progress report on project APRIL. *Journal of Experimental and Theoretical Artificial Intelligence* pp. 271–287.

Theijssen, D., S. Verberne, N. Oostdijk & L. Boves (2007). Evaluating deep syntactic parsing. *Proceedings of the 17th meeting of Computational Linguistics in the Netherlands (CLIN 17)*, Leuven, Belgium, pp. 115–130.

Van Halteren, H. & N. Oostdijk (1993). Towards a syntactic database: The TOSCA analysis system. Aarts, J., P. de Haan & N. Oostdijk (eds.), *English Language Corpora: design, analysis and exploitation*, Rodopi (Amsterdam), pp. 145–161.

Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworths (London).

Verberne, S., L. Boves, N. Oostdijk & P. Coppen (2006). Exploring the use of linguistic analysis for why-question answering. *Proceedings of the 16th meeting of Computational Linguistics in the Netherlands (CLIN 2005)*, Amsterdam, pp. 33–48.