

Chapter 19

Extracting Syntactic Relations using Heuristics

MARK STEVENSON

ABSTRACT. In language processing it is not always necessary to fully parse a text in order to extract the syntactic information needed. We present a shallow parser which extracts a small number of grammatical links between words from unannotated text. The approach used operates by part of speech tagging the text and then applying a set of heuristics. We evaluated our parser by comparing its output against manually identified relations for the same sentences. These results were compared with those reported for a parser constructed by Carroll and Briscoe [Carroll & Briscoe, 1996]. We found that although our results were lower than those reported our system still had a number of advantages: it is computationally far cheaper than full parsing and will process any English sentence.

1 Introduction

Shallow, or partial, parsing has become an important area of natural language processing. This is part of the general movement of language processing from the toy domains and small lexicon used by early systems to large-scale robust modern systems. The practice of creating such systems has become known as *Language Engineering* (LE). An effective LE system should be robust and have wide coverage of the text it is designed to analyse.

We have recently begun research into word sense disambiguation using selectional preferences, an extension of our earlier work ([Stevenson et al., 1998], [Wilks & Stevenson, 1997]). These provide information about the likely sense of the subject, object and indirect object of verbs, the noun modified by an adjective or the verb modified by a verb. In order to apply these restrictions it is necessary to determine the grammatical relations between the words in a sentence. This is usually done by parsing the sentence and extracting the relations forming the resulting tree. However, there are problems with this approach: parsing is computationally expensive and time

consuming, high quality parsers with good linguistic coverage are difficult to find and it is difficult to tell how accurate a parser is at finding these specific relations. We have provided a method which is based on a set of simple heuristics applied to sequences of part of speech tags. This approach is computationally cheap and provides results for any sentence in English.

This paper describes our heuristic parser in detail (Section 2) and report results of a small evaluation we carried out to determine its effectiveness, including a comparison with a well-developed parser constructed by Carroll and Briscoe [Carroll & Briscoe, 1996] (Section 3). We end by drawing some conclusions from our work (Section 4) and suggesting ways in which our parser may be improved (Section 5).

2 The Heuristic Parser

The system developed identifies *simple grammatical links*, binary relations between content words of a sentence which denote grammatical relationships between nouns, verbs, adverbs and adjectives. More specifically, we wish to identify 5 types of relations: ADJ-N, for the adjective modifying a noun, ADV-V for the adverb modifying a verb and SUB-V, OBJ-V, IND-V for the subject, object and indirect object of a verb. For example, for the sentence “The fastest horse won the steeplechase.” we would like to identify these links: ADJ-N between “fastest” and “horse”, SUB-V between “horse” and “won” and OBJ-V between “steeplechase” and “won”.

2.1 The Heuristics

Our system identifies these links by applying a set of simple heuristics to each sentence in a text. Before these heuristics are applied the text is first tokenised, split into sentences and part of speech tagged using Brill’s tagger [Brill, 1992]. The heuristics are then applied to each sentence in the following order:

1. Sentence Segmentation

By manual analysis of example sentences we found that grammatical links do not in general span across certain boundaries and so, if any such boundaries exist in the sentence, we split it into segments which are analysed separately. Segments are defined as quoted phrases within the sentence longer than simply a few words (in the texts we examined it was quite common to find entire sentences quoted within others).

2. Identification of Noun Phrases and Head Nouns

We go through each segment identifying and marking noun phrase

chunks. Part of speech tags are used in this identification. A noun phrase is defined as a maximal sequence consisting of an optional determiner, any number of adjectives (possibly none) and terminated by one or more nouns. The head noun is also marked, this is defined as the rightmost noun in the sequence.

3. Identification of Prepositional Phrases

We then go through each segment and mark prepositional phrases. These are defined as a noun phrase preceded by a preposition. We use the part of speech tags to identify the preposition but, since the tag for preposition (IN) can be assigned to a slightly more general class of words than we would like, we check the preposition against a stop list of prepositions to verify the tag sequence.

4. Identification of ADJ-N and ADV-V relations

Relations between adjectives and nouns and adverbs and verbs are identified using heuristics. A relation between an adjective and noun is postulated for the first noun which is head of a noun phrase and is within four words to the right of the adjective. A relation between an adverb and verb is postulated if there is a verb within two words either side of the adverb and that verb is not an auxiliary verb (a stop list is used to decide which verbs are auxiliaries).

5. Identification of SUB-V, OBJ-V and IND-V relations

For each verb which is not an auxiliary we try to, with identify a subject, object and indirect object. The subject is the first head noun of a noun phrase chunk (not part of a prepositional phrase) to the left of the verb, the direct object the first to the right and the indirect object the second to the right.

It is worth explaining that the identification of verbal arguments does not take the verb's transitivity into account, so if the verb is intransitive (and so has only a subject) the algorithm will still try to identify a direct and indirect object. This does not matter in our application since verbal transitivity is dealt with at a later stage. A more advanced version of this approach could make use of a lexicon containing information on transitivity, hence the number of nouns to look for in relation to the it.

2.2 Worked Example

In order to make the shallow parsing mechanism clearer we provide a short example showing how it would process a particular sentence. The sentence we chose was "The current truancy levels in British schools has reached crisis point." which was taken from the Education Section of the British

Guardian newspaper. The annotations added to the sentence by our parser are printed in **this type**.

The sentence would first have been part of speech tagged. For the sake of clarity we show simpler part of speech tags than those assigned by the part of speech tagger used. The part of speech tagging we would expect for this sentence:

“The/**det** current/**adj** truancy/**noun** levels/**noun** in/**prep** British/**noun** schools/**noun** have/**verb** reached/**verb** crisis/**noun** point/**noun**.”

We now show what each of our heuristics would add to the analysis of this sentence.

1. Sentence segmentation

“The/**det** current/**adj** truancy/**noun** levels/**noun** in/**prep** British/**noun** schools/**noun** have/**verb** reached/**verb** crisis/**noun** point/**noun**.”

The sentence does not contain any bracketed or quoted parts and so it is treated as a single segment.

2. Identification of Noun Phrases and Head Nouns

“[NOUN_PHRASE The/**det** current/**adj** truancy/**noun** [HEAD levels/**noun**]] in/**prep** [NOUN_PHRASE British/**noun** [HEAD schools/**noun**]] have/**verb** reached/**verb** [NOUN_PHRASE crisis/**noun** [HEAD point/**noun**]].”

The second heuristic identifies “The current truancy levels”, “British schools” and “crisis point” as noun phrases, in each case the rightmost noun is chosen as the head noun of the phrase.

3. Identification of Prepositional Phrases

“[NOUN_PHRASE The/**det** current/**adj** truancy/**noun** [HEAD levels/**noun**]] [PREP_PHRASE in/**prep** [NOUN_PHRASE British/**noun** [HEAD schools/**noun**]]] have/**verb** reached/**verb** [NOUN_PHRASE crisis/**noun** [HEAD point/**noun**]].”

The third heuristic identifies “in British schools” as a prepositional phrase since it consists of an appropriate sequence of part of speech tags.

4. Identification of ADJ-N and ADV-V relations

Proposed link: ADJ-N (**current**, **levels**)

The first of the heuristics which identifies relations finds the simple grammatical link between **current** and **levels**. The heuristic chooses the first noun within four words to the right of the adjective, unless it is part of a noun phrase, in which case the head of that phrase is chosen (as has happened in this case).

5. Identification of SUB-V, OBJ-V and IND-V relations

Proposed links: SUB-V (levels, reached), OBJ-V (point, reached)

The final heuristic identifies the arguments of the verbs in the sentence. Since “have” is immediately followed by another verb our system identifies it as an auxiliary and does not propose any links for it. However, it does propose “levels” and “point” as the subject and object of “reached”. Although “schools” is closer to the verb than “levels”, it is not chosen since it is part of a prepositional phrase.

3 Evaluation

We carried out a small evaluation of our approach by comparing our system’s results with simple grammatical links extracted from manually parsed text. We took 25 sentences from the Penn TreeBank [Marcus et al., 1993].¹ We took the parsed versions of these sentences and manually extracted the links we were looking for and gave the raw, unannotated, versions to our system as input. We then compared the links generated by the system with those extracted from the trees by computing recall and precision scores and their combination through the F-measure.² Our system achieved a precision of 51% and recall of 69% over all types of links. Complete results, including results for each type of link computed separately, are shown in Table 19.1. We also counted the number of links generated by our system for each type of link and compared these with the number found in the parsed text, shown in Table 19.2.

Examining the results in Tables 19.1 and 19.2 we see that there is quite a wide difference in our system’s performance over different types of links. There is also a relation between the system’s ability to identify the correct number of links and its accuracy at identifying the link. For example, the two types of link it is most successful at identifying (using F-measure to rank the scores) are ADJ-N and OBJ-V, these are the types of link where the system proposes a very similar number of links to those found in the parsed text.

The system’s worst performance is recorded for IND-V links and this is,

¹These sentences were slightly longer than the average length of 21 word for Penn TreeBank sentences [Gaizauskas, 1995], having an average length of 25 words, the shortest was 5 words long and the longest 41. The sentences, especially the longer ones, contained some grammatical structures of some complexity.

²Recall and precision are complementary evaluation metrics with their roots in Information Retrieval. Recall is defined as the proportion of the instances we would like our system to identify which it actually does. Precision is the proportion of the instances which the system identifies which are valid. The F-measure is a commonly used formula for combining these into a single metric. See [van Rijsbergen, 1979] for further details.

Link Type	Precision	Recall	F-measure
All types	51	69	62
ADJ-N	67	73	71
ADV-V	57	67	63
SUB-V	51	75	65
OBJ-V	63	67	66
IND-V	4	34	10

Table 19.1: Precision, recall and combined scores for link identification

Link Type	System	Parsed	Intersection
ADJ-N	55	51	37
ADV-V	7	6	4
SUB-V	101	68	51
OBJ-V	59	55	37
IND-V	25	3	1
Total	247	183	127

Table 19.2: Number of links identified by system compared to those in parsed text

at least partly, due to the massive over-generation of possible links. As we have already noted (Section 2), this is not a problem since our system simply ignores the proposed indirect object nouns for verbs which are not ditransitive. However, these links cannot be derived from the parsed text and so they have a detrimental effect on our evaluation. (The extra links postulated lower the system’s precision: if we ignore these links and evaluate only over the remaining four types, the system’s precision score increases to 58% and recall remains constant.)

The set of heuristics on which our system is based are extremely simple and only have access to a limited set of knowledge sources (part of speech tags and stop lists). It is very easy to construct examples which would confuse them. However, our evaluation shows that restricted heuristics can still cover many of the constructions which appear in corpora.

3.1 Comparison with Full Parsing

It is unfortunate that few parsers have been quantitatively evaluated, so it is quite difficult to measure how well our system compares to them.

The system which is most similar to ours is Basili et. al.'s shallow parser [Basili et al., 1992], which also identifies binary links between sentence constituents (although they identify a more varied set of links than we attempt to). Unfortunately they have not reported any evaluation of their system. The parser reported by Carroll and Briscoe [Carroll & Briscoe, 1996] has been developed over a number of years, including extensive evaluation and we shall compare our approach with theirs. They use the method suggested by Black [Black et al., 1991] for evaluating parses by computing recall and precision scores of bracket overlap between system generated parses and manually constructed parse trees for the same sentences. Their parser was tested on 250 sentences from the Susanne TreeBank reporting 83% recall and 84% precision. It is not clear how these results relate to the evaluation we carried out and the tests are certainly not identical, however these figures give some idea of the performance that can be expected from a well-developed full parser.

The figures reported by Carroll and Briscoe are significantly higher than those produced by our system (33% higher recall and 14% higher precision) however, their parser only covers 80% of sentences in the test corpus, and was unable to produce any analyses for the remaining 20%.

4 Conclusions

This result shows that computationally simple methods, such as examination of part of speech tag sequences, can be quite effective. Although the results do not appear to be as high as those produced by full parsers the approach described here has the advantage of being more robust and computationally cheaper.

The disadvantage is that our system appears to be less accurate than using a full parse. The usefulness of our results really depends upon the intended application. If a broad coverage approach with reasonable results was desired then a heuristic parsing approach would be useful, especially if computational resources were an issue. However, if we were not interested in broad coverage but wanted a set of syntactic relations which we could be sure were correct then the cost of full parsing may be justified.

5 Further Work

Our approach is quite dependent upon part of speech tags. Although the tagger we use (Brill's Transformation Rule-based Tagger) achieves very good results, it still has an error rate. It would be interesting to determine to what

extent these effect our system's results. As well as being parsed the sentences in the Penn TreeBank are also part of speech tagged with the same syntactic tags used by Brill's tagger. One way to evaluate the error caused by the part of speech tagger would be to compare the results obtained when the correct tags from the Penn TreeBank are used as input to our shallow parser with those obtained when the tagging is carried out automatically. We intend this to be an area of future investigation.

The heuristics developed here are quite specific to English. Another area for future investigation may be to decide how well this approach, with appropriately modified heuristics, would operate on texts from different languages.

Acknowledgements

I would like to thank the European Union who funded the research described here through the Language Engineering project "ECRAN – Extraction of Content: Research at Near-market" (LE-2110).

I am grateful for advice on this research from Mark Hepple and Yorick Wilks and for the comments from two anonymous reviewers of this paper. Any mistakes are my own.

References

- [Basili et al., 1992] R. Basili, M. T. Pazienza, and P. Velardi. A shallow syntactic analyser to extract word associations from corpora. *Literary and Linguistic Computing*, 7(2):114–124, 1992.
- [Black et al., 1991] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorinim, and T. Strzalkowski. A procedure for Quantitively Comparing the Coverage of English. In *Proceedings of the Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA, February 1991. Morgan Kaufmann.
- [Brill, 1992] E. Brill. A simple rule-based part of speech tagger. In *Proceeding of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, 1992.
- [Carroll & Briscoe, 1996] J. Carroll and T. Briscoe. Apportioning Development Effort in a Probabilistic LR Parsing System through Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-96)*, pages 92–100, University of Pennsylvania, May 1996.
- [Gaizauskas, 1995] R. Gaizauskas. Investigations into the grammar underlying the Penn TreeBank II. Research Memorandum CS-95-25, Department of Computer Science, Univeristy of Sheffield, 1995.

- [Marcus et al., 1993] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2):313–330, 1993.
- [Stevenson et al., 1998] M. Stevenson, H. Cunningham, and Y. Wilks. Sense tagging and language engineering. In *European Conference on Artificial Intelligence (ECAI-98)*, pages 185–189, Brighton, UK, 1998.
- [van Rijsbergen, 1979] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.
- [Wilks & Stevenson, 1997] Y. Wilks and M. Stevenson. Sense tagging: Semantic tagging with a lexicon. In *Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics: What, why and how?"*, pages 47–51, Washington, D.C., April 1997. Available as <http://xxx.lanl.gov/ps/cmp-lg/9705016>.