

Context Grammar and POS Tagging

Shian-jung Dick Chen

New Technology and Research
LexisNexis
Ohio, 45342
dick.chen@lexisnexus.com

Don Loritz

New Technology and Research
LexisNexis
Ohio, 45342
don.loritz@lexisnexus.com

Abstract

This paper reports how a rule-based natural language parser uses context knowledge to resolve ambiguities in POS tagging. The parser has only 9 word classes and they are sufficient enough to have fine-grained distinctions and flexible enough to perform their roles in a handful of constituent classes in syntactic parsing. We classify words based on the traditional tripartite of classification – form, function, and distribution (Lyons, 1977). Morphological analysis of form enables the tagger to reduce the number of possible candidate tags for any word to no more than 3, even before parsing. We call it the right context. The words parsed (the left context) provide disambiguation mechanisms with syntactic information that is unavailable to most part-of-speech tagging systems. The POS tagger is highly portable in that there’s no need for data creation or preparation and no training is required for any domain. Also its small tag set gives it the flexibility to tailor its support for other NLP applications.

1 Introduction

This paper reports how FexPars, a natural language parser developed in New Technology & Research of LexisNexis, relies heavily on context knowledge to resolve ambiguities in POS tagging. The parser has only 9 word classes. Before parsing, each word of an input sentence is given a list of no more than four candidate parts of speech (“like” is one of the rare exception of belonging to four classes). The inclusion of POS candidates are based on a word’s senses in the lexicon if it is known and on both derivational and inflectional suffixes, upper or lower cases, and special characters such as numerical or punctuations. The primary goal of POS

tagging is closely tied to the ultimate decision of constituency, attachment, and sense disambiguation in parsing. Based on the traditional tripartite of classification – form, function, and distribution (Lyons, 1977), we let “form” play the role of assigning candidates to unparsed words and leverage “function” and “distribution” as the main contributory factors of context information used for disambiguation.

2 Previous research

Although great success has been reported by some taggers and POS tagging has long been treated as a well-established component technology in NLP, some problems remain outstanding even today. Unknown words still account for a non-negligible portion of the errors reported (Nakagawa et al., 2002). Human-annotated tagging used for training is costly to build (Marcus et al., 1993). Supervised tagging typically suffers from lack of portability, whereas word clusterings resulting from unsupervised fully-automatic methods often miss the fine distinctions found in the carefully designed tag sets used in the supervised systems (Brill, 1992; Schütze, 1993).

POS tagging is rarely used by itself to support a product in IE industry. Products like IBM’s LanguageWare might be the only exception. It’s interesting to note that LanguageWare provides POS for words (mainly from a dictionary) in a document, but it does not care about how customers are going to use them. In other words, POS tagging is regarded as an NLP functionality which has been a must for the past twenty years for anyone claiming doing NLP. Not knowing since when, we notice that POS tagging has become a first yardstick to evaluate NLP systems. The point is, POS

tagging is treated as a stand-alone research effort for a long time and something is wrong with this conception. Treating POS tagging as a stand-alone effort, together with implementing it as a separate process of a large stratificational system from tokenization, to POS tagging and then to parsing, has deprived POS tagging of the rich context information it can use to do the disambiguation task. It is an important objective of this study to advocate full use of context information in POS disambiguation.

We know POS tagging is a well-established research topic and many studies report very good precision numbers doing POS tagging. However, does this mean we all need to direct our efforts elsewhere and there're no significant remaining problems left in POS tagging? It is one of the main objectives of this paper to point out there are still quite a few problems and challenges not adequately addressed by previous researchers. Among them, border effect, linear context inadequacy, multi-functionality, and tagging flexibility are all important issues.

3 Issues and challenges in POS tagging

Here is a list of issues that are not commonly addressed by previous researchers and we also want to point out some challenges that remain challenging even today.

3.1 Issues

- Stratificational v. non-stratificational: Many context-related decisions in the disambiguation process have to wait until parsing. Stratificational systems are doomed to miss something.
- Language creativity and linear context inadequacy: Human language creative power is expected to out-smart any pattern combination mechanism. In other words, there will always be some unseen combinations or patterns.
- Border effect and constituency: Punctuations are not the only phrase or clause delimiters. We expect more intelligence from a system to resolve noun-verb or noun-adjective ambiguity.
- Multi-functional of words: Words are ambiguous in terms of POS because they perform several functions in text.
- Tagging flexibility: Form-based tagging alone does not allow flexible POS change with context.
- Form, function (use) and distribution: The traditional tripartite of classification outlines the requirements for context-based disambiguation.
- Structure, attachment and relation: Structural information enables a system to capture the elements of function and distribution better.
- Unknown words: Their form, function and distribution together help to disambiguate.
- Costs in training corpus building: A larger size of a POS tagger's tagset helps make granular distinction for word classification. However, the large size also makes the construction of training corpus costly.

3.2 Challenges

- Noun-Verb ambiguity and border effect: "a pool of replication-competent HIV remains, which can lead to..." – Is "remains" a Noun or a Verb?
- Adjective-Noun sequence and constituency problem: "Human immunodeficiency viral disease is a chronic, progressive infection." – Is "viral" between two nouns an adjective?
- -ed words and lack of function-distribution information: "The choices popped up as soon as we selected the bore size." – Is "popped" passive or active?
- -ing words and lack of function-distribution information: "The binding results in a conformational change in GP120." – Is "binding" an adjective or a noun? How about "results"?
- "to" – infinitival or PP "a time to place..." – Is "place" a noun or a verb?
- Preposition-subordinate ambiguity:

“never appeared since the train’s schedule...” – Is “since” a preposition or a subordinate conjunction?

- Uppercase, proper names, sentence initials:
“Bush bashing is no longer in their agenda.” – Is “Bush” a common noun or a proper noun?
- numerical and other special characters:
Is it always the case that a numerical character starts a noun phrase?
- comma:
How should comma be classified? How do we know its role (function) in a context?

4 Right context and candidate tags

Our study uses FexPars to make full use of right context and left context. Right context information is provided by means of lexicon lookup, affix categorization and other testing methods. From morphological analysis, any word is given a list of POS tags usually no longer than 3 before parsing, except for rare cases such as “like”, no matter how far away it is from the current position. For example, before parsing the sentence “mRNA coding for adenylate cyclase 1 is weakly increased (~25%) in locus coeruleus upon morphine treatment.”, FexPars provides a list of input words and corresponding POS lists as follows:

```
(mRNA (:n)) (coding (:adj :v :n)) (for (:p :adv)) (adenylate (:n :v :adj)) (cyclase (:unk)) (1 (:n)) (is (:v)) (weakly (:adv)) (increased (:adj :v)) (\ ( (:punc)) (\~ (:punc)) (25 (:n)) (\ ( (:punc)) (in (:p :sub :adv)) (locus (:n)) (coeruleus (:n)) (upon (:p)) (morphine (:n)) (treatment (:n)) (\ ( (:punc))
```

The lists above show that some pre-processing has already filtered candidates for POS tags, even though most words here are not listed in the lexicon. Notice that the largest number of candidate tags in this example is 3. Also notice how derivational/inflectional suffixes and upper case impact the listing of candidate tags.

For words listed in the lexicon, the words derived from different sorts of inflection first undergo morphological analysis to decide their POS because the de-suffixation process find them their stems or roots. For capitalized words, those in some contexts also undergo the de-suffixation

process to find their stems or roots in the lexicon. Other capitalized words will then be assigned Noun as their POS, known (listed in the lexicon) or not. De-capitalization process requires an extra step to do de-suffixation or so when the words are in sentence initial positions or the sentence is one containing all-upper or all-capitalized words.

We have implemented morphological parse for some common derivational inflections. As –s, –ed, and –ing, some derivational inflections are also ambiguous, such as –ant, –al, etc.

At this point, even before syntactic parsing begins, our system has already reduced possible POS candidates to no more than 3 classes for all words of an input sentence. If the sentence is 12 words long and the average number of POS candidates is 2, then the possible combinations are reduced to 4096 in number. Such a pre-processing with the right-context should be very useful even for non-rule-based approach. Nevertheless, with the help of left-context, our system is able to reduce ambiguity to an enormous extent once syntactic parsing begins.

5 Unknown word guessing

The accuracy of part-of-speech tagging for unknown words has often been reported to be substantially lower than that for known words. In FexPars, unknown words, which find no match from lexicon lookup, are prescribed a subset of possible POS tags from a superset of N, V, Adj, and Adv. They are treated not differently from known words of POS ambiguity. In other words, they are disambiguated by context in the same fashion as the processing of known words. The lists given in the previous section show that unknown words with inflectional suffixes will be given a short list of candidate tags, such as (:adv) for –ly ending words, (:adj :adv :n) for –er ending words, (:adj :v :n) for –ing ending words, (:adj :v) for –ed ending words, and (:v :n) for –s ending words. Similar guessing is done for derivational suffixes such as –ion, –ate, –ify. The tag (:unk) is reserved for other non-upper-case, non-numeric, non-hyphenated unknown words. The POS tag of (:unk) means that the unknown word has the longest candidate list (:adv :adj :v :n). Such an arrangement makes unknown word guessing a task

no more complicated than other words listed in the lexicon.

6 Left context and border effect

Left context used in FexPars refers to the known information of a word's parent constituent node and other ancestor nodes. Since POS tagging is integrated into the parsing system, the information derived from parsing always helps to decide if a word is still part of a phrase or clause, or if it starts a new phrase or clause. This is important because the most difficult decisions in POS disambiguation (participles, N-V ambiguity, Adj-N ambiguity and Prep-Sub ambiguity) usually involve some sort of border effect.

Take a word ending with *-ed* for example, it is likely to be a past participle if it is found in a VP following BE or HAVE. Since a word ending with *-ed* can also be a modifier, a cautious test to rule out the possibility of a beginning modifier of another NP is thus suggested before the final decision is made. On the other hand, a past participle can either be a passive or perfective main verb of a matrix clause or a clause-initial participial. The information needed in this case to make a final decision must come from a precise syntactic context, that is, a main verb expected in a clause or a possible phrasal post-modifier of an NP in context. Such context-sensitive information is made available only by the interaction of left context and right context. Such non-linear syntactic information as attachment or embedding is simply not available from an array of neighboring words with diversified morphological features only, no matter how fine-grained the features are.

7 Size of tag set

The size of a tag set, the number of word classes used by a particular system, is a problem not only because there's no easy consensus or not one that fits all, but because it's not easy to find one that works equally well for other NLP tasks beyond POS tagging.

Most state-of-the-art POS taggers tend to populate a tag set to adopt as much fine-grain information as possible, since most systems only have a uniform, linear feature vector as the available context. Take CLAWS POS tagger for in-

stance, its tag set has over 160 tags and it has 47 sub-classes for the traditional class of Noun (Gar-side and Smith, 1997). The good news is that everything ranging from citation to number, proper, upper/lower case, person, grammatical case, temporal, locative, reflexive, etc. are covered. The bad news is that it is expected to take a large proportion of development time to build an annotated corpus big enough to cover all probabilities, given that language is so productive and creative.

8 Use of context information

FexPars is equipped with context knowledge with which the parser is able to make use of context information made available by previous parse history of the left context and the unparsed words of the right context. From the left context, the parser keeps a record of all the parsed words in terms of decided POS, types of constituent to which they belong, the beginning and end of the current constituent, parent and child and sister constituents, and all the useful attributes inferred from the parse. Those pieces of information are stored in the tree structure of the history record according to the node and branch position of each constituent. In other words, the parser uses the tree structure to store context information and fetch them from the corresponding tree structure if any piece of information is needed.

8.1 Example 1 of use of context clues

Take the following sentence for an example: "Due diligence revealed a potential state sales tax charge to the acquiring company in the \$1 million range."

The most difficult decision for the parser to make with respect to POS tagging is in the parsing of the noun phrase "a potential state sales tax charge". Most words in this phrase have at least one part of speech. "Potential" can be noun or adjective, and "state", "tax", and "charge" are both noun and verb. The context knowledge the parse needs most at this point is to be able to decide where the NP should end. In other words, any of the three words qualified in a "verb context" should make the call. However, what is the context knowledge of the parser for a "verb context"?

For one, if there's so far no verb. For two, if the current clause should need a verb, that is, it is

a clausal subject. For three, a qualified verb must agree with the subject in terms of number. Among the three pieces of context knowledge, the second one is the most difficult. It demands the parser to know the type of the current clause, whether it is a subordinate clause, a participial clause, an infinitival clause, or a clause headed by “that” or one of the “Wh-words”.

It turns out the parse so far has already had a verb “revealed”, and the current clause is none of the candidate clause types for a subject clause. As a result, the search for a head noun continues until “sales”. Fortunately, “sales” is one of the rare nouns that end with “-s” and at the same time can be a modifier. So, the search for a head noun goes on until it reaches “to”. Therefore, all the words from “state” to “charge” are all nouns and “charge” is the head noun.

8.2 Example 2 of use of context clues

Now, take the following sentence for another example: “The binding results in a conformational change in GP120.”

The key for successfully parsing this sentence is in deciding that “results” is not a noun but a verb. To make it harder, “results” is the first potential verb of the sentence. But, can we jump to the conclusion that “results” is the one because there isn’t another. Even though the right context allows the parser to get a list of POS candidates, the parser usually do not do this, because potential verbs are just so common.

The previous word “binding” functions more often as a modifier than as a head noun. On the other hand, if “results” is followed by a determiner or the like, it will be more likely to be a verb.

It turns out the best approach is to recognize “result in” as a two-word verb. This example tells us even the use of context clues as such is still not an easy task, not to mention processing with linear context only. Also, we need to be very careful not to over-generate in a case like this.

9 Conclusion

The FexPars used here to do POS tagging takes a rule-based approach. It is not the intention of this paper to favor the rule-based approach to the statis-

tical approach. We know that different approaches have different genius and different limitations. We have show how context knowledge can be used to do POS disambiguation and the task of POS tagging might be better performed if it does not stand alone, left with a choice to use only morphology and linear context.

References

- Eric Brill. 1992. A Simple Rule-based Part of Speech Tagger. *Proceedings of ANLP-92*, 152-155
- Henich Schütze. 1993. Part-of-speech induction from scratch. *Proceedings of the 31st Annual Meeting of ACL*, 251-258.
- John Lyons. 1977. *Semantics*, volume 1. Cambridge University Press. Cambridge, UK.
- M.P. Marcus, B. Santorini and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn tree bank. *Comput. Linguist.* 19:313-330.
- Roger Garside and N. Smith. 1997. A hybrid grammatical tagger: CLAWS4, in Garside, R., Leech, G., and McEnery, A. (eds.) *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Longman, London, 102-121.
- Tetsuji Nakagawa, T. Kudoh, and Y. Matsumoto. 2002. Detecting Errors in Corpora Using Support Vector Machines, *Proceedings of COLING 2002*, 709-715.