An Integrated Statistical Model for Tagging and Chunking Unrestricted Text

Ferran Pla, Antonio Molina, and Natividad Prieto

Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació
Camí de Vera s/n 46020 València
{fpla,amolina,nprieto}@dsic.upv.es

Abstract. In this paper, we present a corpus-based approach for tagging and chunking. The formalism used is based on stochastic finite-state automata. Therefore, it can include n-grams models or any stochastic finite-state automata learnt using grammatical inference techniques. As the models involved in our system are learnt automatically, it allows for a very flexible and portable system for different languages and chunk definitions. In order to show the viability of our approach, we present results for tagging and chunking using different combinations of bigrams and other more complex automata learnt by means of the Error Correcting Grammatical Inference (ECGI) algorithm. The experimentation was carried out on the Wall Street Journal corpus for English and on the LexEsp corpus for Spanish.

1 Introduction

Part of Speech Tagging and Shallow Parsing are two well-known problems in Natural Language Processing. A tagger can be considered as a translator that reads sentences from a certain language and outputs the corresponding sequences of part-of-speech (POS) tags, taking into account the context in which each word of the sentence appears. A Shallow Parser involves dividing sentences into non-overlapping segments on the basis of a very superficial analysis. It includes discovering the main constituents of the sentences (NPs, VPs, PPs,...) and their heads. Shallow Parsing usually identifies non-recursive constituents, also called chunks (such as non-recursive Noun Phrases or base NP, base VP, and so on) [1]. Shallow Parsing is used as a fast and reliable pre-processing phase for full or partial parsing. It can be used for Information Retrieval Systems, Information Extraction, Text Summarization, Bilingual Alignment, etc. The different approaches for solving Tagging problem can be classified into two main groups, depending on the tendencies followed for establishing the Language Model: the linguistic approach, which is based on hand-coded linguistic rules [6], [22], and the learning approach which is derived from a corpora (labelled or non-labelled), using different formalisms: HMM [8], [15], Decision Trees [11], [13], Maximum Entropy [20]. Other approximations that use hybrid methods have also been proposed [23]. Shallow parsing techniques can also be classified into the same two groups as above. These approaches have a common characteristic: they take the sequence of lexical tags proposed by a POS tagger as input for the chunking process. Most linguistic approaches use finite state methods for detecting chunks or for accomplishing other linguistic tasks [2], [4], [12]. Other works use different grammatical formalisms (such as constraint grammars) [21], or combine the grammar rules with a set of heuristics [5]. Learning technique approaches automatically construct a language model from a labelled and bracketed corpus. In [8], a stochastic model for detecting simple noun phrases is learnt. Transformation-based learning was used in [19] to detect base NP. The Memory-Based Learning algorithm [10] takes into account lexical and POS information. The Memory-Based Sequence Learning algorithm [3] learns substrings or sequences of POS and brackets.

2 General Description of our System for Tagging and Chunking

In this work, we present an integrated system that combines different knowledge sources (lexical probabilities, models for chunks and a contextual model for the sentences) for tagging and chunking texts from a certain language. The approach is based on stochastic finite-state models that are learnt automatically, so we achieved a very flexible and portable system. The models that we have used are based on bigrams and other finite-state automata which were learnt using grammatical inference techniques [17], [18]. Our system can be considered as a two-level transducer. The upper one describes contextual information about the structure of the sentences, and the lower one modelizes the structure of the chunks considered.

All these models have been estimated from labelled and bracketed corpora. The training set is composed by sentences which are marked with a begin label and an end label for each chunk, and each word is labelled with its corresponding part-of-speech tag. Once the different models have been learnt, a regular substitution of the lower models into the upper one is made. In this way, we get a single integrated model which shows the possible concatenations of lexical tags and syntactical units. This integrated model includes the transition probabilities as well as the lexical probabilities.

The lexical probabilities are estimated from the word frequencies, the tag frequencies and the word-per-tag frequencies. For English, we used a tag dictionary which was built from the entire corpus. It gives us all the possible lexical categories (POS tags) for each word; this is equivalent to having an ideal morphological analyzer. For the Spanish task, we have incorporated the morphological analyzer MACO [7]. In both cases, the probabilities for each possible tag were assigned from this information taking into account the obtained statistics. Due to the fact that a word may not have been seen at training, or it may have only been seen in some of the possible categories (not all), it is necessary to apply a smoothing mechanism. In our approach, if the word has not previously been seen, the same probability is assigned to all the categories given by the dictionary; if it has been seen, but not in all the categories, the smoothing mechanism called "add one" is applied. Afterwards, a renormalization process is carried out.

The tagging and shallow parsing process consists of finding out the sequence of states of maximum probability on the integrated model for an input sentence. Therefore, this sequence must be compatible with the contextual, syntactical and lexical constraints. This process can be carried out by Dynamic Programming using the Viterbi algorithm, which we modified to adapt to our models. From the Dynamic Programming trellis, we can not only obtain the maximum probability path for the input sentence through the model, and the best sequence of lexical tags but also the best segmentation into chunks.

3 Experimental Work

In order to evaluate the approach proposed here, we conducted some experiments to estimate the tagging accuracy rate and the precision and recall rates for NP-chunk detection. The models that represent contextual information and NP-chunks structure were learnt from training data using the SLM-toolkit developed by CMU [9] or using the ECGI algorithm. In the first case, we used smoothed bigrams (BIG), and in the second one, we used the so-called ECGI automata which was also smoothed. The experiments were carried out on the WSJ corpus, using the lexical tags defined in [14], taking only the NP chunks defined by [8] in account. Nevertheless, the use of this approach on other corpora (changing the reference language), other lexical tag sets or other kinds of chunks could be done in a direct way. In particular, we conducted some experiments on the LexEsp Spanish Corpus [7] using a different tag set, and a different chunks definition.

For the experiments, we used 900,000 words out of the entire WSJ corpus (800,000 words for training and 100,000 words for testing). In Table 1, we show the results for tagging and NP-chunking on the test set. Each row in the table corresponds to a certain kind of model for the upper level and for the lower level. For example, the first row (BIG-BIG) shows the results using the integrated model when we used bigrams in order to modelize the two levels; the second one (BIG-ECGI) corresponds to an integrated model where we used a bigram to modelize the contextual information and the ECGI automata to describe the structure of NP chunks, and so on.

Table 1. Tagging and NP-chunking on WSJ corpus. Results using the two-level transducer (800 kwords for training and 100 kwords for testing)

Method	NP-Precision	NP-Recall	Tagging Accuracy
BIG-BIG	94.55%	93.60%	96.76%
BIG-ECGI	93.74%	93.09%	96.66%
ECGI-ECGI	93.17%	91.44%	96.56%
ECGI-BIG	93.75%	91.80%	96.61%

The best results for tagging were obtained on BIG-BIG models (96.8%), This value was slightly lower than the obtained using a simple bigram model (96.9%) and was the same using a simple ECGI automaton. The results obtained for NP chunking were very satisfactory achieving a precision rate of 94.5% and a recall rate of 93.6%.

We conducted similar experiments on the Spanish Corpus LexEsp, using a smaller data set: 70,000 words for training and 25,000 words for testing. The results are presented in Table 2. In this case, we obtained a tagging accuracy rate of 96.9% using BIG-BIG or BIG-ECGI integrated models. When we used a simple bigram model, we obtained a rate of 97.0% (96.8% using a single ECGI automaton). The results obtained for NP-chunking achieved a precision rate of 93.2% and a recall rate of 92.7%.

When we worked on simpler tasks (simple syntax and reduced vocabulary), the approach based on ECGI was a bit better. For instance, over the Spanish corpus BDGEO [16], the tagging accuracy was 99.2% using ECGI models and 99.0% using BIG.

Table 2. Tagging and NP-chunking on the Spanish Lexesp corpus. Results using the two-level transducer (70 kwords for training and 25 kwords for testing)

Method	NP-Precision	NP-Recall	Tagging Accuracy
BIG-BIG	93.18%	92.74%	96.92%
BIG-ECGI	92.46%	91.97%	96.92%
ECGI-ECGI	91.82%	91.42%	96.80%
ECGI-BIG	91.79%	91.52%	96.74%

Even though the results presented using ECGI automata were, in general, slightly worse than those obtained using only bigrams, more work should be done in order to further develop all the capabilities of the ECGI approach. We are working currently in order to introduce certain adjustment factors between the probability distributions involved in the process.

4 Conclusions and Future Work

The proposed framework constitutes an attractive approach for tagging and shallow parsing in a single process. It is based on stochastic finite-state automata that are learnt automatically from data. It allows for a very flexible and portable system.

The comparison of results among different approaches proposed by other authors is difficult due to the multiple factors that must be considered: the language, the number and kinds of tags, the size of the vocabulary, the ambiguity, the difficulty of the test set, etc. Nevertheless we believe that the results reported here are competitive (96.9% for part-of-speech tagging accuracy and a 94.9% of precision rate of NP-chunking, on the WSJ corpus).

In addition, the usual sequential process for chunking a sentence can also be used. That is, first we tag the sentence (using a single bigram model or other available tagger) and then we use the integrated model to carry out the chunking from the output of the tagger. In this case, only the contextual models are taken into account in the decoding process (without lexical probabilities). The performance of this sequential process slightly improves the recall rate (94.1%). We think that this is due to the way we combined the probabilities of the different models.

In general, tagging and chunking results obtained using ECGI models are worse than those obtained using bigrams. We have observed that the differences in the tagging errors are around 3%, even if the accuracy is the same. This fact encourages us to deeply study the kind of errors in order to characterize them. We are working on the possibility of combining this information in order to increase the performance of the system.

Also, the system can easily incorporate the contextual information of the words using structural tags. That is, we can specialize certain part-of-speech tags using lexical information. Preliminary results obtained using this approach show that precision and recall rates for certain kinds of chunks are improved.

Acknowledgements

This work has been supported by the Spanish Research Project TIC97-0671-C02-01/02

References

1. S. Abney. *Parsing by Chunks.* R. Berwick, S. Abney and C. Tenny (eds.) Principle–based Parsing . Kluwer Academic Publishers, Dordrecht, 1991.
2. S. Abney. Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI'96 Robust Parsing Workshop*, Prague, Czech Republic, 1996.
3. S. Argamon, I. Dagan, and Y. Krymolowski. A Memory–based Approach to Learning Shallow Natural Language Patterns. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 67–73, Montréal, Canada, 1998.
4. S. At-Mokhtar and J. Chanod. Incremental Finite-State Parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington D.C., USA, 1997.
5. D. Bourigault. Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 977–981, 1992.

6. E. Brill. Transformation–based Error–driven Learning and Natural Language Processing: A Case Study in Part–of–speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.
7. J. Carmona, S. Cervell, L. Màrquez, M. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 915–922, Granada, Spain, May 1998.
8. K. W. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the 1st Conference on Applied Natural Language Processing, ANLP*, pages 136–143. ACL, 1988.
9. P. Clarksond and R. Ronsenfeld. Statistical Language Modeling using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.
10. W. Daelemans, S. Buchholz, and J. Veenstra. Memory-Based Shallow Parsing. In *Proceedings of EMNLP/VLC-99*, pages 239–246, University of Maryland, USA, June 1999.
11. W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. MBT: A Memory–Based Part–of–speech Tagger Generator. In *Proceedings of the 4th Workshop on Very Large Corpora*, pages 14–27, Copenhagen, Denmark, 1996.
12. E. Ejerhed. Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods . In *Proceedings of Second Conference on Applied Natural Language Processing*, pages 219–227. ACL, 1988.
13. D. M. Magerman. Learning Grammatical Structure Using Statistical Decision–Trees. In *Proceedings of the 3rd International Colloquium on Grammatical Inference, ICGI*, pages 1–21, 1996. Springer-Verlag Lecture Notes Series in Artificial Intelligence 1147.
14. M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 1993.
15. B. Merialdo. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171, 1994.
16. F. Pla and A. Molina. Etiquetado Morfosintáctico del Corpus BDGEO. In *Proceedings of the CAEPIA*, Murcia, España, November 1999.
17. F. Pla and N. Prieto. Using Grammatical Inference Methods for Automatic Part–of–speech Tagging. In *Proceedings of 1st International Conference on Language Resources and Evaluation, LREC*, Granada, Spain, 1998.
18. N. Prieto and E. Vidal. Learning Language Models through the ECGI Method. *Speech Communication*, 1:299–309, 1992.
19. L. Ramshaw and M. Marcus. Text Chunking Using Transformation-Based Learning. In *Proceedings of third Workshop on Very Large Corpora*, pages 82–94, June 1995.
20. A. Ratnaparkhi. A Maximum Entropy Part–of–speech Tagger. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*, 1996.
21. A. Voutilainen. NPTool, a Detector of English Noun Phrases. In *Proceedings of the Workshop on Very Large Corpora*. ACL, June 1993.
22. A. Voutilainen. A Syntax–Based Part–of–speech Analyzer. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, Dublin, Ireland, 1995.
23. A. Voutilainen and L. Padró. Developing a Hybrid NP Parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, pages 80–87, Washington DC, 1997. ACL.