N-Best Extraction for Weighted Tree Automata*

Anna Jonsson

Department of Computing Science, Umeå University, 90187 Umeå, Sweden
aj@cs.umu.se

Handling natural language is central to the development of AI: a computer cannot pass the Turing test without a well-functioning natural language system. In natural language processing, we often face the problem of having to choose between a large, or even infinite, number of hypotheses. We investigate the problem of finding the top N hypotheses given a hypothesis space in which every element is assigned a fitness value, probability, or the like. This value is usually called a *weight*, and the problem of finding the top N hypotheses is known as the *N-best problem*. Solving this problem is important to, e.g., reduce the amount of data handled in certain applications. Let us consider the case of machine translation.

Machine translation can be done in the form of a cascade consisting of several steps for which the output of the previous step becomes the input of the next one [3]. The intermediate data then consists of a set of sentences-in-progress. If we were to propagate all of the intermediate data in each step, we would get a combinatorial explosion. Reducing the data in the least result-affecting way possible is thus important for the efficiency and accuracy of such software.

Since solving the N-best problem is very dependent on the type of hypothesis space, we cannot solve the general problem, but we need to narrow it down. One way of doing this is to say that the weights have to be assigned to the hypotheses by a formal device, such as a probabilistic grammar. Such grammars are important components in recent attempts to marry traditional discrete models of language with neural networks [6].

If we in the machine translation example represent the sentences as strings, and the weights are assigned to them using a weighted finite automaton, then we can solve the N-best problem by applying the algorithm presented in [4]. Our contribution is generalising this N-best-strings algorithm to trees to allow for the use of syntax trees and thus grammatical devices such as the probabilistic grammars mentioned above. The idea behind our solution is to look at just enough trees to be sure which one should be output next. We present our solution in the form of the algorithm Best Trees to which the input is a weighted tree automaton (wta) whose weights can be seen as error scores.

To evaluate Best Trees, we run it on real machine translation data resulting from [5]. This allows us to compare it to the state-of-the-art for extracting the N best trees, namely the algorithm presented in [1]. Simply speaking, this algorithm finds the N best trees, though – as opposed to Best Trees – possibly including duplicates. It is implemented in the tree automata toolkit Tiburon [2]. We slightly modified the Tiburon code such that it only outputs trees that it has

* Supported by the Swedish Research Council, Grant No. 621-2012-4555.

not seen before, which allowed us to run TIBURON on the same data, and thereby compare it to BEST TREES with respect to the N-best-trees problem.

Preliminary results indicate that BEST TREES performs slightly better than TIBURON on this particular data (see Fig. 1). To find a more solid conclusion, however, we need to further analyse the data, but we can for certain say that BEST TREES is at least as good as the current state-of-the-art.

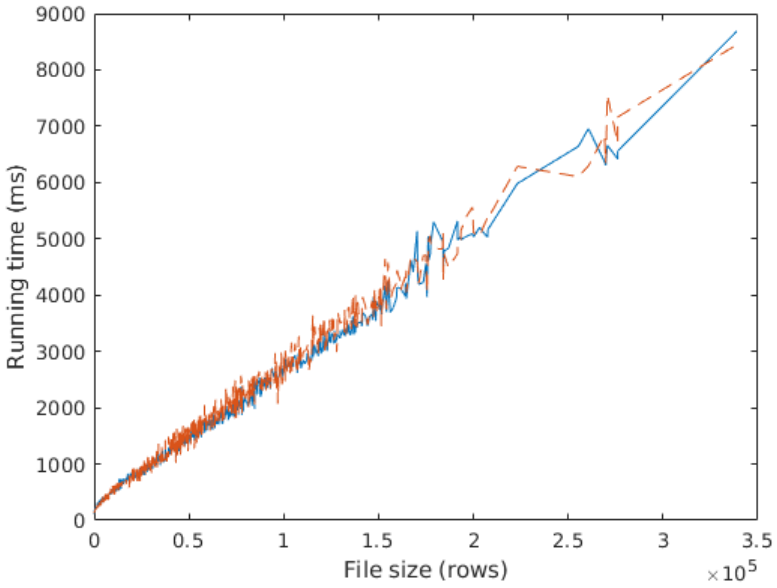

Fig. 1. Running times of BEST TREES (solid) and TIBURON (dashed) on the machine translation data set consisting of different-sized wtas. Each data point is the mean value of the running times for $N = 10, 20, ..., 200$ (the result for each value of N is a mean of ten runs).

References

1. Huang, L., Chiang, D.: Better k-best parsing. In: Proc. of the Conference on Parsing Technology 2005. pp. 53–64. Association for Computational Linguistics (2005)
2. May, J., Knight, K.: Tiburon: A weighted tree automata toolkit. In: International Conference on Impl. and Application of Automata. pp. 102–113. Springer (2006)
3. May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1058–1066 (2010)
4. Mohri, M., Riley, M.: An efficient algorithm for the n-best-strings problem. In: Proceedings of the Conference on Spoken Language Processing (2002)
5. Quernheim, D.: Bimorphism Machine Translation. Ph.D. thesis (2017)
6. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 455–465 (2013)