proceedings

MDPI

Extended Abstract

Increasing NLP Parsing Efficiency with Chunking †

Mark Dáibhidh Anderson * and David Vilares

FASTPARSE Lab, Departamento de Computación, University of A Coruña, Campus de Elviña, 15071 A Coruña, Spain; david.vilares@udc.es

* Correspondence: m.anderson@udc.es; Tel.: +34-981-167-000

† Presented at the XoveTIC Congress, A Coruña, Spain, 27–28 September 2018.

Published: 19 September 2018

Abstract: We introduce a "Chunk-and-Pass" parsing technique influenced by a psycholinguistic model, where linguistic information is processed not word-by-word but rather in larger chunks of words. We present preliminary results that show that it is feasible to compress linguistic data into chunks without significantly diminishing parsing performance and potentially increasing the speed.

Keywords: Parsing; Syntax; natural language processing; NLP; dependency parsing; Chunking

1. Introduction

Syntactic information is required to fully understand linguistic information: utterances are not just a string of words with a meaning solely derived from the semantics of each individual word. The way they are combined also affects meaning. In this context, syntactic analysis can augment many applications in natural language processing (NLP), e.g., state-of-the-art semantic analysis and information retrieval. Dependency parsers are used in these systems as the other main flavour of parsers, constituency parsers, are orders of magnitude slower. Still, state-of-the-art dependency parsers can only process about 100 sentences per second [1]. For large-scale analyses, this is cost-prohibitive.

Dependency parsing represents relations between words with arcs, e.g., the phrase "*I felt*" would have an arc from "*felt*" (the head) to "*I*" (the dependent) and with a *nsubj* label (see Figure 1). Attachment scores are used to evaluate dependency parsers: unlabelled (UAS) measures the number of correct heads and labelled (LAS) measures this and the labelling accuracy.

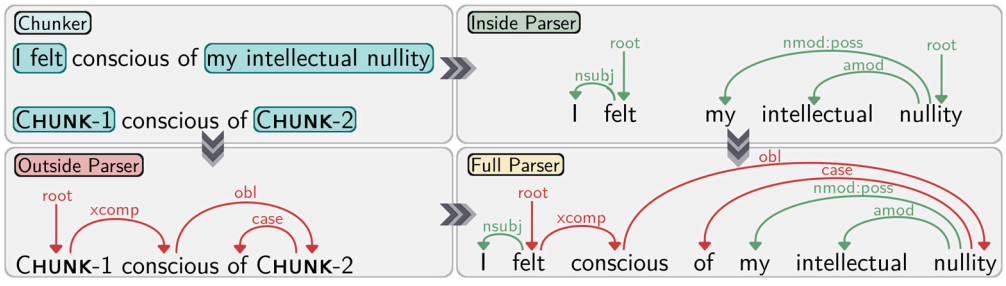

Figure 1. Initially the sentence is processed by the chunker. The contents of the chunks are then sent to the inside parser and the abstract representation of the sentence is sent to the outside parser. The predictions from both are then collated to form the full parse.

Our technique to increase parsing efficiency is inspired by the "Chunk - and - Pass" psycholinguistic model, where an ever increasing abstract hierarchical representation of linguistic input is created in order to process it efficiently and to overcome working-memory restrictions [2]. This entails finding phrases in sentences which can be extracted and processed by a faster but less

robust parser, while the more abstract form with more complicated relations is parsed by a slower and more thorough parser.

2. Materials and Methods

The implementation consists of a supervised chunker; an inside parser which analyses the words within the chunks; and an outside parser which analyses the relationships between chunks (see Figure 1). The dataset used was the Universal Dependency English EWT treebank v2.1 [3].

The supervised chunker was implemented using a neural sequence labelling toolkit (NCRF++) [4]. We generated gold labels for the chunker using the BIO tagging scheme, where B is the beginning, I is inside, and O is outside of a chunk. B and I tags were suffixed with the phrase type of the chunk, e.g., B-NP and I-NP for noun-phrase chunks. The labels were generated by using part-of-speech rule sets automatically extracted from the training data. An example rule for a noun phrase could be DET ADJ NOUN. Each set has a threshold on the ratio between invalid (containing unrelated words) and valid chunks when used with an unsupervised rule-based chunker.

The inside parser used the arc eager algorithm in MaltParser [5]. The outside parser used a neural network (NN) implementation of the stack-based arc standard algorithm [6] with universal-dependency-specific features [7]. The inside parser has a speed of ≈16,500 tokens per second (TPS), the chunker of ≈10,200 TPS, and the outside of ≈2000 TPS, so a compression ratio (initial tokens to resulting chunks) of 1.6 can theoretically increase the speed relative to using just the NN parser by 15%.

3. Results

Figure 2a shows the dependency of the supervised chunker's performance on the global ratio threshold of the rule sets used to generate gold-labelled data as described above. Also in Figure 2a the chunker's compression ratio with respect to the rule threshold is shown. Figure 2b shows the parsing performance of the full system, the inside parser, the outside parser, and the corresponding performance of the baseline model (NN stack-based arc standard) for each.

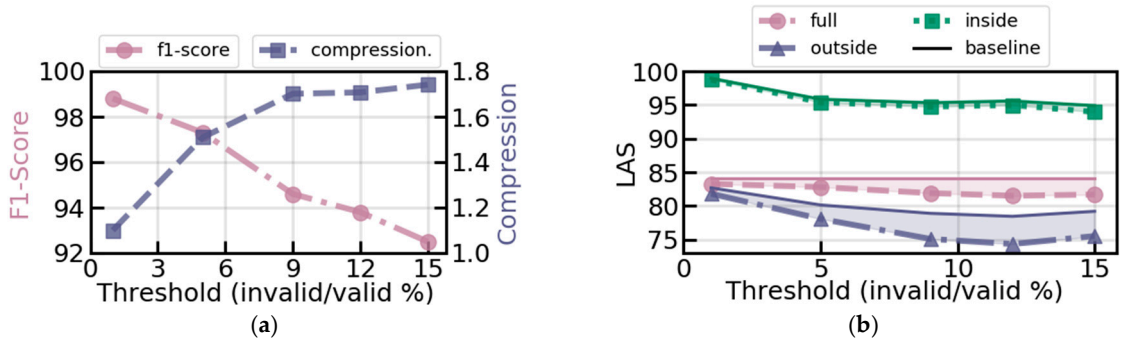

Figure 2. (**a**) NCRF++ performance and the corresponding compression rate for different rule sets. (**b**) Inside (green, square), outside (blue, triangle), and full-system (magenta, circle) scores using NCRF++ chunker for different rule sets. Baseline refers to the performances of the baseline model for the corresponding sections that were sent to each sub-parser and are displayed as continuous lines.

4. Discussion

As seen in Figure 2a, it is not useful to use rule sets with ever decreasing performances as the compression return begins to diminish, so there appears to be an upper limit of efficiency improvement. In Figure 2b, it can be observed that the inside chunker does not lose much accuracy. The loss is more pronounced for the outside parser. This is likely due to the decrease in contextual information it has and the more complicated relationships it has to process. Despite this, the best compression to performance rule set (9% threshold) only loses 1.25 UAS and 2.2 LAS points.

We have shown initial results that highlight the efficacy of this approach. Further research will be focused on optimising the implementation and acquiring accurate speed measurements. Beyond this, we will expand the system to process other languages.

Funding: This work has received funding from the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150).

References

1. Gómez-Rodríguez, C. Towards fast natural language parsing: FASTPARSE ERC Starting Grant. *Proces. Leng. Nat.* **2017**, *59*, 121–124.
2. Christiansen, M.H.; Chater, N. The Now-or-Never bottleneck: A fundamental constraint on language. *Behav. Brain Sci.* **2016**, *39*, e62, doi:10.1017/S0140525X1500031X.
3. Nivre, J.; Agić, Ž.; Ahrenberg, L.; Antonsen, L.; Aranzabe, M.J.; Asahara, M.; Ateyah, L.; Attia, M.; Atutxa, A.; Augustinus, L.; et al. *Universal Dependencies 2.1*; LINDAT/CLARIN Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University: Prague, Czech Republic, 2017.
4. Yang, J.; Zhang, Y. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018.
5. Nivre, J.; Hall, J.; Nilsson, J. Maltparser: A data-driven parser-generator for dependency parsing. In Proceedings of LREC, Genoa, Italy, May 2006; pp. 2216–2219.
6. Chen, D.; Manning, C. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014; pp. 740–750.
7. Straka, M.; Hajic, J.; Straková, J.; Hajic jr, J. Parsing universal dependency treebanks using neural networks and search-based oracle. In Proceedings of the International Workshop on Treebanks and Linguistic Theories (TLT14), Warsaw, Poland, 11–12 December 2015; pp. 208–220.