An Efficient, Generic Approach to Extracting Multi-Word Expressions from Dependency Trees

Scott Martens and Vincent Vandeghinste
Centrum voor Computerlinguïstiek
Katholieke Universiteit Leuven
scott@ccl.kuleuven.be & vincent@ccl.kuleuven.be

Abstract

The *Varro* toolkit offers an intuitive mechanism for extracting *syntactically motivated* multi-word expressions (MWEs) from dependency treebanks by looking for recurring connected subtrees instead of subsequences in strings. This approach can find MWEs that are in varying orders and have words inserted into their components. This paper also proposes *description length gain* as a statistical correlation measure well-suited to tree structures.

1 Introduction

Automatic MWE extraction techniques operate by using either statistical correlation tests on the distributions of words in corpora, syntactic pattern matching techniques, or by using hypotheses about the semantic non-compositionality of MWEs. This paper proposes a purely statistical technique for MWE extraction that incorporates syntactic considerations by operating entirely on dependency treebanks. On the whole, dependency trees have one node for each word in the sentence, although most dependency schemes vary from this to some extent in practice. See Figure 1 for an example dependency tree produced automatically by the Stanford parser from the English language data in the *Europarl corpus*. (Marneffe, 2008; Koehn, 2005)

Identifying MWEs with subtrees in dependency trees is not a new idea. It is close to the formal definition offered in Mel'čuk (1998), and is applied computationally in Debusmann (2004) However, using dependency treebanks to automatically extract MWEs is fairly new and few MWE extraction projects to date take advantage of dependency information directly. There are a number of reasons why this is the case:

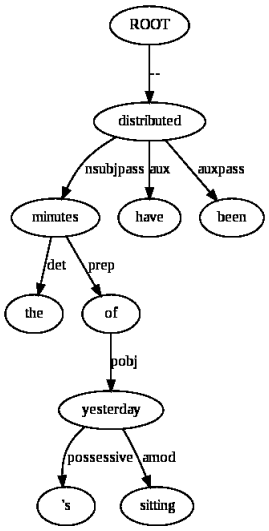
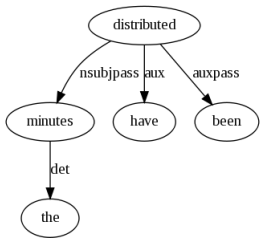


Figure 1. A *dependency tree* of the sentence *"The Minutes of yesterday's sitting have been distributed."*

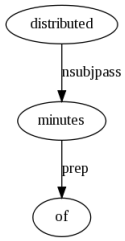
- String-based algorithms are not readily applicable to trees.
- Tree structures yield a potentially combinatorial number of candidate MWEs, a problem shared with methods that look for strings with gaps.
- Statistical techniques used in MWE extraction, like *pointwise mutual information*, are two-variable tests that are not easy to apply to larger sets of words.

The tool and statistical procedures used in this research are not language dependent and can operate on MWE of *any size*, producing depen-

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 85–88,
Beijing, August 2010



(a) "The Minutes (...) have been distributed"



(b) "(...) Minutes of (...) distributed."

Figure 2. Two induced subtrees of the dependency tree in Figure 1. Note that both correspond to discontinuous phrases in the original sentence.

dency pairs, short phrases of any syntactic category, lengthy formulas and idioms. There are no underlying linguistic assumptions in this methodology except that *a MWE must consist of words that have a fixed set of dependency links in a treebank.* Even word order and distance between words is not directly assumed to be significant. The input, however, requires substantial linguistic pre-processing – particularly, the identification of at least some of the dependency relations in the corpora used. Retrieving MWEs that contain abstract categories, like information about the arguments of verbs or part-of-speech information for unincluded elements, requires using treebanks that contain that information, rather than purely lexical dependency trees.

2 Varro Toolkit for Frequent Subtree Discovery

The Varro toolkit is an open-source application for efficiently extracting *frequent closed unordered induced subtrees* from treebanks with labeled nodes and edges. It is publicly available under an open source license.[1] For a fuller description of *Varro*, including the algorithm and data structures used and a formal definition of *frequent closed unordered induced subtrees*, see Martens (2010).

Given some tree like the one in Figure 1, an *induced subtree* is a connected subset of its nodes and the edges that connect them, as shown in Figure 2. Subtrees do not necessarily represent fixed sequences of words in the original text, they include syntactically motivated discontinuous phrases. This dramatically reduces the number of candidate discontinuous MWEs when compared to string methods. An *unordered induced subtree* is a subtree where the words may appear with different word orders, but the subtree is still identified as the same if the dependency structure is the same. A *frequent closed subtree* is a subtree of a treebank that appears more than some fixed number of times and where there is no subtree that contains it and appears the same number of times. Finding only *closed* subtrees reduces the combinatorial explosion of possible subtrees, and ensures that each candidate MWE includes all the words the that co-occur with it every time it appears.

3 Preprocessing and Extracting Subtrees

The English language portion of the *Europarl Corpus, version 3* was parsed using the Stanford parser, which produces both a constituentcy parse and a dependency tree as its output.[2] The dependency information for each sentence was transformed into the XML input format used by *Varro*. The result is a treebank of 1.4 million individual parse trees, each representing a sentence, and a total of 36 million nodes.

In order to test the suitability of *Varro* for large treebanks and intensive extractions, all recurring closed subtrees that appear at least *twice* were extracted. This took a total of 129,312.27 seconds (just over 34 hours), producing 9,976,355 frequent subtrees, of which 9,909,269 contain more than one word and are therefore candidate MWEs.

A fragment of the *Varro* output can be seen in Figure 3. The nodes of the subtrees returned are not in a grammatical surface order. However, the original source order can be recovered by using the locations where each subtree appears to find the order in the treebank. Doing so for the tree in Figure 3 shows what kinds of MWEs this approach can extract from treebanks. The underlined words in the following sentences are the ones included in the subtree in Figure 3:

[1] http://varro.sourceforge.net/

[2] This portion of the work was done by our colleagues Jörg Tiedemann and Gideon Kotzé at RU Groningen.

```
<subtree rootCount="2581"
         entropy="97.2382532056"
         dlg="194892.881978"
         mi="75.510609058"
         compression="0.776552504478" >
  <tree>
    <node edge="root" label="ROOT" >
     <node edge="--" label="take" >
      <node edge="nsubj" label="vote" >
       <node edge="det" label="the" />
      </node>
      <node edge="dobj" label="place" />
      <node edge="prep" label="at" />
      <node edge="aux" label="will" />
     </node>
    </node>
  </tree>
  <addresses>
    <node id="ep-05-02-22:649:0" />
    <node id="ep-05-02-22:2712:0" />
    <node id="ep-05-02-22:2981:0" />
    <node id="ep-05-02-22:3126:0" />
    <node id="ep-05-02-22:3204:0" />
    <node id="ep-05-02-22:3420:0" />
    <node id="ep-99-01-14:1510:0" />
    <node id="ep-99-01-14:1595:0" />
    <node id="ep-99-01-14:3075:0" />
    <node id="ep-01-01-18:392:0" />
    <node id="ep-01-01-18:1091:0" />
```

Figure 3. An example of a found subtree and candidate MWE. This subtree appears in 2581 unique locations in the treebank, and only the locations of the first few places in the treebank where it appears are reproduced here, but all 2581 are in the *Varro* output data.

The vote will take place tomorrow at 9 a.m.
The vote will take place today at noon.
The vote will take place tomorrow, Wednesday at 11:30 a.m.

4 Statistical Methods for Evaluating Subtrees as MWEs

To evaluate the quality of subtrees as MWEs, we propose to use a simplified form of *description length gain* (DLG), a metric derived from algorithmic information theory and Minimum Description Length methods (MDL). (Rissanen, 1978; Grünwald, 2005) Given a quantity of data of any kind that can be stored as a digital information in a computer, and some process which transforms the data in a way that can be reversed, DLG is the measure of how the space required to store that data changes when it is transformed.

To calculate DLG, one must first decide how to encode the trees in the treebank. It is not necessary to actually encode the treebank in any particular format. All that is necessary is to be able to calculate how many bits the treebank would require to encode it.

Space prevents the full description of the encoding mechanism used or the way DLG is calculated. The encoding mechanism is largely the same as the one described in Luccio et al. (2001) Converting the trees to strings makes it possible to calculate the encoding size by calculating the entropy of the treebank in that encoding using classical information theoric methods.

In effect, the procedure for calculating DLG is to calculate the entropy of the whole treebank, given the encoding method chosen, and then to recalculate its entropy given some subtree which is removed from the treebank and replaced with a symbol that acts as an abbreviation. That subtree is then be added back to the treebank once as part of a look-up table. These methods are largely the same as those used by common data compression software.

DLG is the difference between these two entropy measures.[3]

Because of the sensitivity of DLG to low frequencies, it can be viewed as a kind of non-parametric significance test. Any frequent structure that cannot be used to compress the treebank has a negative DLG and is not frequent enough or large enough to be considered significant.

Varro reports several statistics related to DLG for each extracted subtree, as shown in Figure 3:

- Unique appearances (reported by the *rootCount* attribute) is the number of times the extracted subtree appears with a different root node.
- *Entropy* is the entropy of the extracted subtree, given the encoding scheme that *Varro* uses to calculate DLG.
- *Algorithmic mutual information* (AMI) (reported with the *mi* attribute) is the DLG of the extracted subtree divided by its number of unique appearances in the treebank.
- *Compression* is the AMI divided by the entropy.

AMI is comparable to *pointwise mutual information* (PMI) in that both are measures of redundant bits, while *compression* is comparable to *normalized mutual information* metrics.

[3]This is a *very simplified* picture of MDL and DLG metrics.

5 Results and Conclusions

We used the metrics described above to sort the nearly 10 million frequent subtrees of the parsed English Europarl corpus. We found that:

- *Compression* and AMI metrics strongly favor very large subtrees that represent highly formulaic language.
- *DLG* alone finds smaller, high frequency expressions more like MWEs favoured by terminologists and collocation analysis.

For example, the highest DLG subtree matches the phrase *"the European Union"*. This is not unexpected given the source of the data and constitutes a very positive result. Among the nearly 10 million candidate MWEs extracted, it also places near the top discontinuous phrases like *"... am speaking ... in my ... capacity as ..."*.

Using both compression ratio and AMI, the same subtree appears first. It is present 26 times in the treebank, with a compression score of 0.894 and an AMI of 386.92 bits. It corresponds to the underlined words in the sentence below:

> The next item is the recommendation for second reading (A4-0245/99), on behalf of the Committee on Transport and Tourism, on the common position adopted by the Council (13651/3/98 - C4-0037/99-96/0182 (COD) with a view to adopting a Council Directive on the charging of heavy goods vehicles for the use of certain infrastructures.

This is precisely the kind of formulaic speech, with various gaps to fill in, which is of great interest for *sub-sentential translation memory systems*. (Gotti et al., 2005; Vandeghinste and Martens, 2010)

We believe this kind of strategy can substantially enhance MWE extraction techniques. It integrates syntax into MWE extraction in an intuitive way. Furthermore, description length gain offers a unified statistical account of an MWE as a linguistically motivated structure that can compress relevant corpus data. It is similar to the types of statistical tests already used, but is also non-parametric and suitable for the study of arbitrary MWEs, not just two-word MWEs or phrases that occur without gaps.

6 Acknowledgements

This research is supported by the AMASS++ Project,[4] directly funded by the *Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT)* (SBO IWT 060051) and by the PaCo-MT project (STE-07007).

References

Debusmann, Ralph. 2004. Multiword expressions as dependency subgraphs. *Proceedings of the 2004 ACL Workshop on Multiword Expressions*, pp. 56-63.

Gotti, Fabrizio, Philippe Langlais, Eliott Macklovitch, Didier Bourigault, Benoit Robichaud and Claude Coulombe. 2005. 3GTM: A third-generation translation memory. *Proceedings of the 3rd Computational Linguistics in the North-East Workshop*, pp. 8–15.

Grünwald, Peter. 2005. A tutorial introduction to the minimum description length principle. In: *Advances in Minimum Description Length: Theory and Applications*, (Peter Grünwald, In Jae Myung, Mark Pitt, eds.), MIT Press, pp. 23–81.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. Proceedings of the 10th MT Summit, pp. 79–86.

Luccio, Fabrizio, Antonio Enriquez, Pablo Rieumont and Linda Pagli. 2001. *Exact Rooted Subtree Matching in Sublinear Time*. Università di Pisa Technical Report TR-01-14.

de Marneffe, Marie-Catherine and Christopher D. Manning. 2008. The Stanford typed dependencies representation. *Proceedings of the 2008 CoLing Workshop on Cross-framework and Cross-domain Parser Evaluation*, pp. 1–8.

Martens, Scott. 2010. Varro: An Algorithm and Toolkit for Regular Structure Discovery in Treebanks. *Proceedings of the 2010 Int'l Conf. on Computational Linguistics (CoLing)*, in press.

Mel'čuk, Igor. 1998. Collocations and Lexical Functions. In: *Phraseology. Theory, Analysis, and Applications*, (Anthony Cowie ed.), pp. 23–53.

Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatica*, vol. 14, pp. 465–471.

Vandeghinste, Vincent and Scott Martens. 2010. Bottom-up transfer in Example-based Machine Translation. *Proceedings of the 2010 Conf. of the European Association for Machine Translation*, in press.

[4] http://www.cs.kuleuven.be/~liir/projects/amass/