Poly-GrETEL: Cross-Lingual Example-based Querying of Syntactic Constructions

Liesbeth Augustinus, Vincent Vandeghinste, Tom Vanallemeersch
Centre for Computational Linguistics
KU Leuven (University of Leuven), Leuven, Belgium
{firstname.lastname}@kuleuven.be

Abstract

We present Poly-GrETEL, an online tool which enables syntactic querying in parallel treebanks and which is based on the monolingual GrETEL environment. We provide online access to the Europarl parallel treebank for Dutch and English, allowing users to query the treebank using either an XPath expression or an example sentence in order to look for similar constructions. We provide automatic alignments between the nodes. By combining example-based query functionality with node alignments, we limit the need for users to be familiar with the query language and the structure of the trees in the source and target language, thus facilitating the use of parallel corpora for comparative linguistics and translation studies.

Keywords: parallel treebanks, example-based querying, search tool, alignment

1. Introduction

In this paper we describe Poly-GrETEL,[1] the continued development of GrETEL (Greedy Extraction of Trees for Empirical Linguistics), that was started as an online query engine for a manually verified, syntactically annotated corpus of written Dutch (1 million words) (Augustinus et al., 2012), allowing to look for syntactic matches using examples. These examples are automatically converted into an XPath expression,[2] which is used to query the treebank. Example-based querying has the advantage that users do not need to be familiar with the query language, nor with the lay-out of the underlying XML structure used to represent the trees, nor with the grammar used by the parser. Alternatively, users can directly query the treebank using XPath, without examples.

GrETEL has further been extended with a 1-million word, manually verified treebank of spoken Dutch (Augustinus et al., 2013), and the interface has been improved. We also added a large (automatically annotated) treebank of 500 million words for querying by example (Vandeghinste and Augustinus, 2014).

We now further expand the scope of GrETEL by adding a treebank for Dutch, i.e. the automatically parsed Europarl treebank, and its parallel counterpart for English, extracted from the Europarl parallel corpus (Koehn, 2005), version 7. We refer to this extended engine as Poly-GrETEL.

By making parallel data easily queryable and freely available online, we allow translators, language learners, and people working in comparative linguistics and translation studies to look up how data are translated in reality. Instead of restricting search to string matching, we allow to check how certain syntactic patterns are actually translated. When looking at lexical information in combination with syntactic patterns, the engine functions as a *bilingual syntactic concordancer*.

[1] http://gretel.ccl.kuleuven.be/poly-gretel

[2] http://www.w3.org/TR/xpath

2. Related Work

A parallel corpus consists of pairs of source sentences and their translation. It is a specific instance of a multilingual corpus, which may contain sentences in more than two languages. Parallel corpora are useful for comparative linguistics, translation studies, language learning and creation of machine translation systems. Many corpora with two or more languages are currently available, one of the most popular being Europarl (Koehn, 2005). An example of a more recently constructed parallel corpus resulted from the Indian Languages Corpora Initiative (ILCI), a project on the construction of parallel annotated corpora for 17 Indian languages, including English (Bansal et al., 2013).

Parallel corpora may consist of raw data or may be enriched with linguistic annotations, such as lemmas and part-of-speech tags.

A number of parallel *treebanks* have been built, which provide manually verified parse trees for the sentences in both languages. For instance, the SMULTRON treebank (Volk et al., 2015) contains parses for languages like English, German and Swedish. The ParGramBank environment (Sulger et al., 2013) contains LFG parses for typologically diverse languages.

Alignment information in parallel corpora comes in the form of links between words, constituents or dependencies. For instance, the SMULTRON treebank contains handcrafted alignments between nodes in parse trees. The PaCo-MT system (Vandeghinste et al., 2013) and the SCATE project (Vandeghinste et al., 2015) make use of large sets of *automatically* created parse trees and subtree alignment links to induce synchronous grammars for machine translation.

Depending on the available linguistic annotation and alignment, and on the expressiveness of the query language, the parallel corpus may be queried in different ways. The OPUS environment (Tiedemann, 2012) uses the Corpus Workbench (Evert and Hardie, 2011) for querying with regular expressions and part-of-speech tags. The Stockholm TreeAligner (Lundborg et al., 2007) and the INESS Search

platform (Meurer, 2012) allow querying parse tree nodes, e.g. by specifying the type of constituent.
Queries typically relate to the source language only. Some query tools allow for a cross-language search, i.e. the specification of constraints in both the source and the target language. The Stockholm TreeAligner and the INESS Search platform add an alignment condition to this cross-language search: parse pairs satisfying the query are only shown if the matching parse nodes are aligned (Volk et al., 2014).
Formulating cross-language queries on trees is complex, as it usually requires knowledge of the query language and of the structure of the source and target language trees. By using *cross-lingual example-based querying*, Poly-GrETEL avoids this complexity.

3. The parallel treebank

We extracted the Dutch and English sentences from the Europarl parallel corpus (Koehn, 2005), version 7.[3] Table 1 presents statistics about the corpus.

	Words	Sentences
Dutch	38,859,141	1,607,423
English	40,077,179	1,607,423

Table 1: Statistics about the Europarl parallel corpus

We parsed the Dutch side with Alpino (van Noord, 2006), a dependency parser that also assigns phrase structure labels and outputs XML trees (*Alpino-XML format*) that are isomorphous to the syntax trees, which makes them easily queryable with XPath.
We parsed the English side using the Stanford parser (Klein and Manning, 2003), added the dependency labels (de Marneffe et al., 2006), and converted the bracketed phrase-structure tree and the dependency labels into Alpino-XML format.
We added word alignment information by applying GIZA++ (Och and Ney, 2003), as well as node alignment between parallel trees provided by the Dublin Subtree Aligner (Zhechev, 2009), which creates node alignments in a relatively straightforward manner, using lexical probabilities derived by GIZA++. Nodes adhering to well-formedness rules are aligned. An alternative alignment, which we intend to include in future versions, is produced by Lingua-Align (Tiedemann, 2010), a discriminative tree aligner trained on a small parallel treebank with manual alignments.

4. Querying the treebank

Previous implementations of GrETEL allow users to look up Dutch constructions, either by example or using XPath queries.
When lookup is example-based, the input example is parsed using the same parser that is used for the creation of the treebank, and the user indicates the relevant and irrelevant parts in the parse tree of the example. This information is automatically converted into XPath, which is used for the actual treebank search.
In Poly-GrETEL, users can not only query the Dutch side and find English equivalents, but they can also query the English side in an example-based mode, or using XPath directly (monolingual search). Alternatively, the user can combine Dutch and English examples into one query (bilingual search).
Figure 1 presents the architecture of Poly-GrETEL. Section 4.1. illustrates the bilingual example-based search mode, whereas section 4.2. presents the example-based search mode for monolingual queries. In section 4.3. the XPath search mode will be discussed and compared to example-based querying.

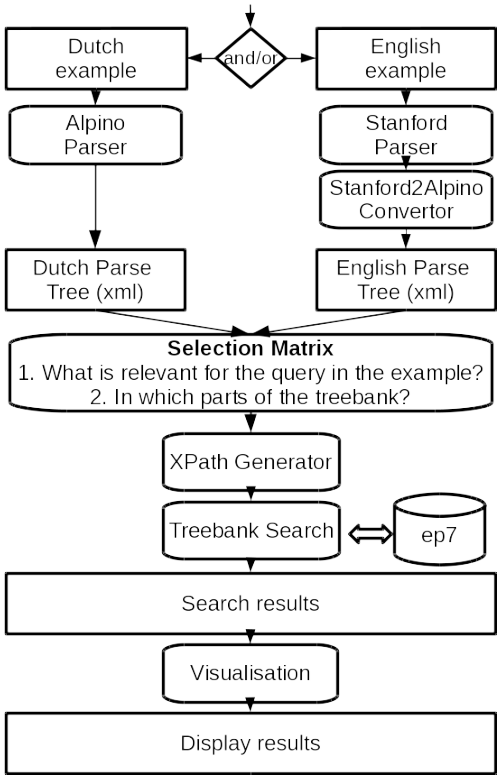

Figure 1: Flow chart of the processing steps for example-based querying in Poly-GrETEL

4.1. Bilingual search

1. Bilingual example The user provides two examples containing the syntactic constructions under investigation. In this example we look for translations of a VP into a deverbal nominalisation. We could for instance specify the English example in (1) and the Dutch example in (2). The words marked in bold are the ones relevant for the constructions we are interested in.

(1) It is difficult to **reach** the **top**.

(2) **Het bereiken van** de **top** is moeilijk.
the reach.INF of the top is difficult
'Reaching the top is difficult.'

2. Parse Poly-GrETEL automatically parses the Dutch input construction with the Alpino parser and the English

[3] The corpus was downloaded from `http://www.statmt.org/europarl`.

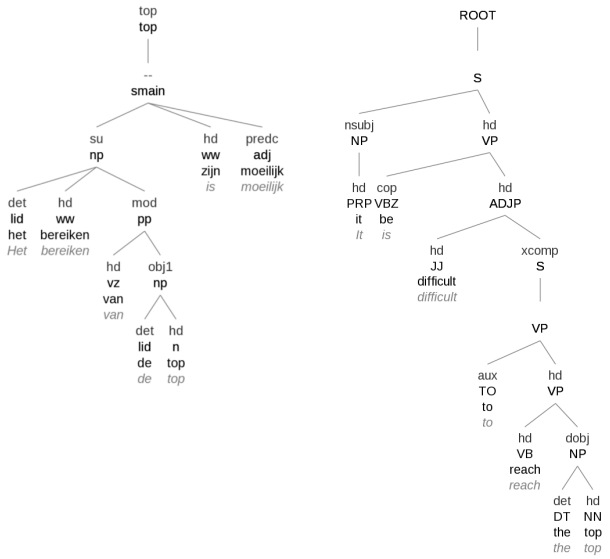

Figure 2: Alpino (left) and Stanford (right) parses of the input sentences

construction using the Stanford parser. The resulting parses are presented to the user as syntax trees, cf. Figure 2.

3. Selection matrix Poly-GrETEL returns the input examples in a matrix, in which the user indicates the parts of the construction that are relevant for the search, as well as the level of generalization. For each word in the construction the user can indicate whether the word class, lemma or exact word form is part of the query. Words that are irrelevant for the query but were provided as context for the parsers should be indicated as 'optional in search'. Figure 3 shows the selection matrices for both the Dutch and English construction.

Figure 3: Selection matrix of the Dutch (top) and English (bottom) construction

In the example, we are looking for non-lexical constructions: Only word class is indicated for the VP *reach (the) top* in the English construction, and the deverbal nominalisation *het bereiken van (de) top* 'reaching the top' in the Dutch construction.

In the bilingual search mode the checkbox 'align subsentential constructions' is checked by default.[4] This means that Poly-GrETEL will look for matches in which the constructions are aligned. If the user unchecks this box, GrETEL will look for constructions in which both constructions occur in the same translation unit, but are not necessarily aligned.

4. Treebank selection In this step the user can choose in which treebanks the constructions should be looked up. The Europarl parallel treebank is split up in subtreebanks per year. It is possible to select one or more components. For this example, the Europarl component of the year 2000 was chosen.

5. Query This step presents an overview of the query. Based on the parses of the examples and the information in the matrices, Poly-GrETEL extracts the subtrees containing the construction under investigation from the parse trees, as illustrated in Figure 4.

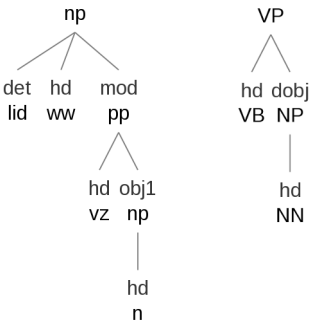

Figure 4: Subtrees based on the parses of the input sentences and the information provided by the user

The subtrees in Figure 4 serve as input to the XPath generator, which automatically generates the corresponding XPath expressions, used to query the parallel treebank. The XPath query matching the English construction is presented in (3).

(3)
```
//node[@cat="VP"and
node[@rel="hd" and @pos="VB"] and
node[@rel="dobj" and @cat="NP" and
node[@rel="hd" and @pos="NN"]]]
```

In the basic search mode, the queries are not shown at this stage. In the advanced search mode, users can adapt the XPath expressions in order to refine or generalize the search instruction. If a pair of parse trees matches the Dutch and English query, Poly-GrETEL checks whether the matching parts are aligned.

[4]For more information about the two other search options, see the GrETEL manual (`http://gretel.ccl.kuleuven.be/project`) and Augustinus et al. (2012).

6. Results If Poly-GrETEL finds matching constructions in the parallel treebank, they are presented to the user as a list of sentence pairs. Some example results are given in (4–5). The a-sentences are matches for the source side and the b-sentences are the corresponding construction in the target side. The parts in bold are the aligned matches.

(4) a. The reasons for adjustments are, for example, **to improve employment**...

b. De redenen voor de aanpassingen zijn bijvoorbeeld **het verbeteren van de werkgelegenheid**...

(5) a. The Commission agrees to **strengthen this political message**.

b. De Commissie gaat akkoord met **het versterken van deze politieke boodschap**.

Poly-GrETEL allows for visualizing parses and node-aligned matches. The alignment of parts of the sentences in (5) is shown in Figure 5.[5] For the sake of clarity, the top part of the parse trees is not shown in the figure, and adjacent aligned nodes are marked in bold. For instance, in the English group *VB – VP – NP*, all three nodes are aligned (*strengthen* is aligned to *versterken*, etc.), but only one link is drawn between the group and the corresponding group in Dutch.

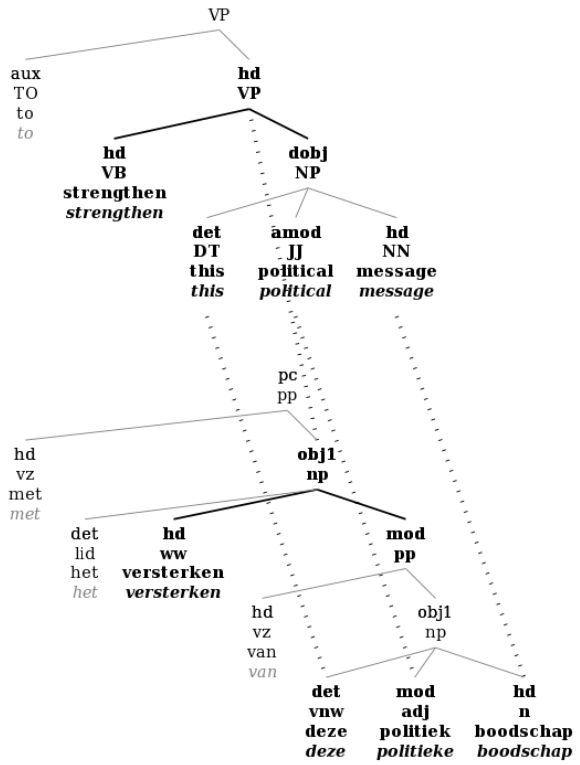

Figure 5: Visualization of aligned matching parts in a search result

The search procedure illustrated in this section presented the novelty of example-based searching in parallel treebanks, i.e. defining constraints for both source and target constructions using examples, and requiring they are aligned. One can also perform a parallel search without the requirement that both constructions are aligned. Another alternative is to query only one side of the treebank, i.e. monolingual search.

[5]The alignment was produced by the Dublin Subtree Aligner.

4.2. Monolingual search

This section illustrates the example-based search mode for a monolingual query. We start from an English example, which means that GrETEL skips the processing steps for Dutch (see Figure 1).

1. Monolingual example In this example we look for translations of English copular constructions in which the subject is a gerund containing a direct object, e.g. (6).

(6) Querying a treebank is easy.

2. Parse In this step only one parse is returned, cf. Figure 6. The parser used depends on the language of the input construction. In this case the input is parsed using the Stanford parser.

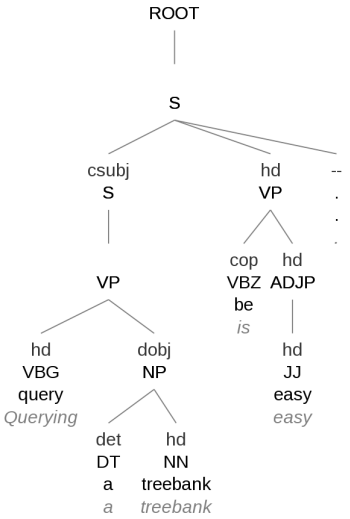

Figure 6: Parse of the input sentence

3. Selection matrix Analogous to the bilingual example-based search, the input example is returned in a matrix, cf. Figure 7. As only one example is entered, only one matrix is shown, and it is not possible to specify alignment constraints.

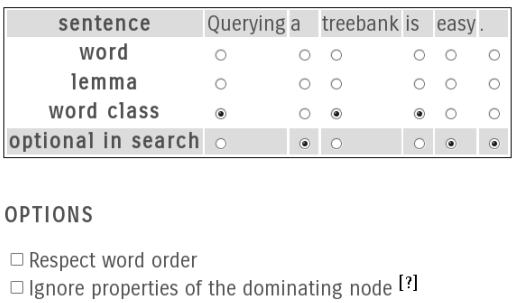

Figure 7: Selection matrix

In order to find similar constructions, the gerund *querying*, its object *treebank*, and the verb *is* are indicated as relevant. The verb is indicated in order to include the dependency relation of the gerund in the query tree (cf. step 5), i.e. the requirement that it is the subject of the sentence.

4. Treebank selection This step is identical to treebank selection in the bilingual search mode. For this example we have chosen the Europarl component of 2001.

5. Query Based on the input sentence and the user input provided in the selection matrix, Poly-GrETEL extracts the subtree in Figure 8.

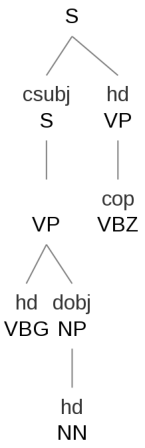

Figure 8: Query tree

The automatically generated XPath instruction corresponding to the query tree in Figure 8 is given in (7).

(7)
```
//node[@cat="S" and
node[@rel="csubj" and @cat="S"
and node[@cat="VP" and
node[@rel="hd" and @pos="VBG"]
and node[@rel="dobj" and
@cat="NP" and node[@rel="hd" and
@pos="NN"]]]] and node[@rel="hd"
and @cat="VP" and node[@rel="cop"
and @pos="VBZ"]]]
```

6. Results Again the results are presented as a list of matching sentence pairs. Some examples are given in (8–9). The search instruction only matches constructions in the source side (a-sentences), but Poly-GrETEL also presents the translations of the matching sentences (b-sentences).

(8) a. **Drafting a European constitution is** another perverse fantasy.

b. Het opstellen van een Europese constitutie is een ander waandenkbeeld.

(9) a. **Maintaining access for this humanitarian aid is** crucial ...

b. Het is van cruciaal belang dat de toegang voor deze humanitaire hulp behouden blijft ...

4.3. XPath search

Example-based search has many advantages. For instance, the user only needs a limited knowledge of XPath, the annotation guidelines and the lay-out of the XML structure in which the treebank is encoded.

Still, XPath querying enhances the query flexibility compared to the example-based approach. Therefore, another approach of querying the corpora in GrETEL consists of directly formulating an XPath query describing the syntactic pattern the user is looking for. This query is then processed in the same way as the automatically generated query in the example-based approach.

As mentioned in section 4.1. it is possible to manually adapt the generated XPath query in the advanced search mode of example-based querying before querying the treebank. This can be seen as an intermediate approach, as an XPath query is easier to understand and adapt than to construct from scratch.

For instance, the results of the query in (7) all contain a form of the copula, as the verb used for the input example was the copula, which is indicated by the dependency relation `cop`. If one wants to generalize over all verb forms, the query in (7) can be adapted to the one in (10).

(10)
```
//node[@cat="S" and
node[@rel="csubj" and @cat="S"
and node[@cat="VP" and
node[@rel="hd" and @pos="VBG"] and
node[@rel="dobj" and @cat="NP" and
node[@rel="hd" and @pos="NN"]]]]
and node[@rel="hd" and @cat="VP"
and node[@pos="VBZ"]]]
```

In addition to the constructions in (8–9), the results of the query in (10) also include the constructions in (11–12), as the dependency relation of the verb is underspecified in the adapted query.

(11) a. For me, **reducing debt means** taking responsibility for the future.

b. Voor mij betekent schuldenvermindering verantwoordelijkheid voor de toekomst nemen.

(12) a. **Increasing marine safety requires** ongoing work.

b. Het vergroten van de veiligheid op zee vereist voortdurende maatregelen.

5. Conclusions and future work

We have presented Poly-GrETEL, an online search engine to query parallel treebanks, providing access to the Europarl parallel treebank (Dutch-English).

Future work includes the creation of parallel treebanks containing other languages. We will also provide filtering mechanisms based on the alignment probabilities and/or the number of alignments in the matching subtree. We will add more generalized POS tags to the English part of the treebank, in order to allow formulation of English search instructions at a similar abstraction level as the Dutch queries. Furthermore, we aim to speed up the treebank search by applying the preprocessing methodology described in Vandeghinste and Augustinus (2014) to the Europarl treebank.

Acknowledgements

Poly-GrETEL is developed in the context of the SCATE project (Smart Computer-Aided Translation Environment),[6] funded by the Flemish Agency for Innovation through Science and Technology (IWT SBO, Project Nr. 130041).

6. Bibliographical References

Augustinus, L., Vandeghinste, V., and Van Eynde, F. (2012). Example-Based Treebank Querying. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3161–3167, Istanbul, Turkey.

Augustinus, L., Vandeghinste, V., Schuurman, I., and Van Eynde, F. (2013). Example-Based Treebank Querying with GrETEL – now also for Spoken Dutch. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013). NEALT Proceedings Series 16*, pages 423–428, Oslo, Norway.

Bansal, A., Banerjee, E., and Nath Jha, G. (2013). Corpora Creation for Indian Language Technologies - The ILCI Project. In *Proceedings of the 6th Language & Technology Conference (LTC'13)*, pages 253–257, Poznań, Poland.

de Marneffe, M., MacCartney, B., and Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454, Genoa, Italy.

Evert, S. and Hardie, A. (2011). Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 Conference*, Birmingham, UK.

Klein, D. and Manning, C. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of ACL*, pages 423–430, Sapporo, Japan.

Lundborg, J., Marek, T., Mettler, M., and Volk, M. (2007). Using the Stockholm TreeAligner. In *The 6th International Workshop on Treebanks and Linguistic Theories (TLT 6)*, pages 73–78, Bergen, Norway.

Meurer, P. (2012). INESS-Search: a search system for LFG (and other) treebanks. In *Proceedings of 17th International LFG Conference*, pages 404–421.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Tiedemann, J. (2010). Lingua-Align: An Experimental Toolbox for Automatic Tree-to-Tree Alignment. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valetta, Malta.

van Noord, G. (2006). At Last Parsing Is Now Operational. In *TALN 2006. Verbum Ex Machina. Actes de la 13ème Conference sur le Traitement Automatique des Langues Naturelles*, pages 20–42.

Vandeghinste, V. and Augustinus, L. (2014). Making Large Treebanks Searchable. The SoNaR case. In *Proceedings of the LREC 2014 2nd workshop on Challenges in the management of large corpora (CMLC-2)*, pages 15–20, Reykjavik, Iceland.

Vandeghinste, V., Martens, S., Kotzé, G., Tiedemann, J., Bogaert, J. V. D., Smet, K. D., Eynde, F. V., and van Noord, G. (2013). Parse and Corpus-based Machine Translation. In Peter Spyns et al., editors, *Essential Speech and Language Technology for Dutch: resources, tools and applications*, chapter 17, pages 305–319. Springer.

Vandeghinste, V., Vanallemeersch, T., Eynde, F. V., Heyman, G., Moens, S., Pelemans, J., Wambacq, P., der Lek-Ciudin, I. V., Tezcan, A., Macken, L., Hoste, V., Geurts, E., and Haesen, M. (2015). Smart Computer Aided Translation Environment. In *Proceedings of the 18th Annual Conference of the European Association for Machine Tranlation (EAMT 2015)*, Antalya, Turkey.

Volk, M., Graën, J., and Callegaro, E. (2014). Innovations in Parallel Corpus Search Tools. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 3172–3178, Reykjavik, Iceland, May.

Zhechev, V. (2009). *Automatic generation of parallel treebanks: an efficient unsupervised system.* Ph.D. thesis, Dublin City University.

7. Language Resource References

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit X*, pages 79–86.

Sulger, S., Butt, M., King, T. H., Meurer, P., Laczkó, T., Rákosi, G., Dione, C. B., Dyvik, H., Rosén, V., De Smedt, K., Patejuk, A., Cetinoglu, O., Arka, I. W., and Mistica, M. (2013). ParGramBank: The ParGram Parallel Treebank. In *Proceedings of the 51st Annual Meeting of ACL*, pages 550–560, Sofia, Bulgaria.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2214–2218, Istanbul, Turkey.

Volk, M., Ghring, A., Rios, A., Marek, T., and Samuelsson, Y. (2015). SMULTRON (version 4.0) – The Stockholm MULtilingual parallel TReebank.

[6] http://www.ccl.kuleuven.be/scate