DECCA Project Description

Detmar Meurers and Markus Dickinson

1 Introduction

In the past decade, research and applications in human language technology have strongly been influenced by the success of data-driven and stochastic modeling of natural language based on electronic corpora annotated with linguistic information. Annotated corpora are fundamental for training and testing algorithms in statistical natural language processing, and they are essential as gold standards for testing the performance of human language technology. The annotations in such corpora are treated as accurate even though it is known that the linguistic annotations contain a significant number of errors—and researchers have started to point out the inadequacy of such corpora for evaluating and for training human language technology (Padro and Marquez 1998; van Halteren et al. 2001). The purpose of this project is to explore a new method for automatically detecting and correcting errors in corpora, to extend its applicability and its precision and recall for different types of linguistic annotation, and to investigate its relation to fraud/anomaly detection approaches developed outside of computational linguistics.

In a corpus, consistency of annotation clearly is desirable, but because high-quality annotations of relevant linguistic properties cannot be obtained fully automatically (with current technology), humans are involved in the correction or annotation process, so that inconsistencies arise. Humans do not always agree; they may be inconsistent; they may not fully understand the guidelines; and they are subject to factors such as fatigue or apathy. Thus, manually-corrected or annotated corpora are likely to contain variation in the annotation, and this variation frequently is erroneous.

The fact that variation in annotation can indicate annotation errors has recently been used to detect errors in annotation (van Halteren 2000; Nakagawa and Matsumoto 2002). Our research project develops error detection via variation into a general method which automatically detects variation in annotation within recurring contexts. We have prototyped this method on part-of-speech annotations (Dickinson and Meurers 2003a) and demonstrated the feasibility of extending it to cover syntactic annotation using traditional tree structures (Dickinson and Meurers 2003b) or potentially discontinuous structures (Dickinson and Meurers 2005a). Widening the range of labeled data to which the approach is applicable, we will investigate its application to linguistic dependency structures and explore the relation of our method to approaches used for anomaly detection outside of computational linguistics, e.g., in medical diagnosis validation (Gamberger et al. 1996). The project will explore increasing the recall of the error detection method by generalizing the notion of a recurring context and investigate the connection to machine learning methods in order to extend the approach from error detection to error correction. Finally, to research the practical relevance of detecting and correcting the annotation errors, it will determine the effect of correcting the detected errors on natural language technology trained on those resources.

2 The approach: Error detection via variation

Our error detection research (Dickinson and Meurers 2003a,b, 2005a,b) takes the idea that variation in corpus annotation can be an indicator of annotation errors as a point of departure to develop a method which automatically detects varying part-of-speech and syntactic annotation within recurring contexts.

2.1 Using the variation in a corpus

Part-of-speech (POS) annotation provides a good starting point to see how variation can occur in a corpus. The POS tagging process reduces a set of lexically possible tags to the correct tag for a specific corpus occurrence. A particular word occurring more than once in a corpus can thus be assigned different tags. This is what in Dickinson and Meurers (2003a) we refer to as *variation*, and it is caused by one of two reasons: i) *ambiguity*: there is a word ("type") with multiple possible tags and different corpus occurrences of that word ("tokens") happen to realize the different options,or ii) *error*: the tagging of a word is inconsistent across comparable occurrences.

The more similar the context of a variation, the more likely it is for the variation to be an error. In our research, we have focused on contexts composed of words, and we require identity of the context. We use the term *variation n-gram* for an n-gram (of words) in a corpus that contains a word annotated differently in another occurrence of the same n-gram in the corpus. The word exhibiting the variation is referred to as the *variation nucleus.* We exploit the fact that every variation n-gram contains two variation (n–1)-grams to calculate the variation n-grams efficiently, using an instance of the a priori algorithm (Agrawal and Srikant 1994).

Given that there are many different types of linguistic annotation (as well as a wide range of labeled data outside of linguistics), we want to explore what classes of annotated data our method can be generalized to. In the following, we exemplify why such extensions of the variation detection method are needed and how they can proceed.

Detecting variation in syntactic annotation We introduced variation n-grams for annotation with a one-to-one mapping between the tokens and the annotation, such as POS annotation. Structural annotation does not meet this criterion directly, but in Dickinson and Meurers (2003b) we show how the variation detection method can be extended to annotation labeling strings of tokens, such as syntactic annotation. In order to maintain a theory-independent, data-driven definition of the nuclei for which the annotation is compared, we decompose the variation n-gram detection for syntactic annotation into a series of runs with different nucleus sizes, thereby establishing a one-to-one relation between a unit of data and a syntactic category annotation. Each run detects the variation in the annotation of strings of a specific length. By performing such runs for strings from length 1 to the length of the longest constituent in the corpus, we ensure that all strings which are analyzed as a constituent somewhere in the corpus are compared to the annotation of other occurrences of that string.

Detecting variation in discontinuous syntactic annotation For languages with relatively free constituent order, such as German, Dutch or the Slavic languages, the combinatorial potential of the language encoded in constituency cannot be mapped straightforwardly onto the word order possibilities of those languages. As a consequence, the treebanks that have been created for German (NEGRA, Skut et al. 1997; Verbmobil, Hinrichs et al. 2000; TIGER, Brants et al. 2002) have relaxed the requirement that constituents have to be contiguous.

Discontinuous constituents present a serious challenge to an error detection method based on the assumption that a contiguous string of a particular length can be mapped to a single category. In order to extend the method to discontinuous constituents, in Dickinson and Meurers (2005a) we developed a technique which is capable of comparing labels for any set of corpus positions, instead of for any interval. In searching for variation nuclei, we have to be able to find both discontinuous constituents and discontinuous non-constituents which match those constituents. To find such non-constituents, we make use of a trie data structure (Fredkin 1960) to store constituents and then attempt to match strings in the corpus with a path in the trie.

2.2 Heuristics for classifying variation

Once the variation n-grams for a corpus have been computed, heuristics are used to classify the variations into errors and ambiguities. The first heuristic used by our method encodes the basic fact that the label assignment for a nucleus is dependent on the context of that nucleus: variation in longer n-grams are more likely to be errors. The second takes into account that natural languages favor the use of local dependencies over non-local ones: variation nuclei on the fringe of an n-gram are more likely to be genuine ambiguities.

2.3 Results of the approach so far

We tested the variation error detection method on the Wall Street Journal (WSJ) corpus, part of the Penn Treebank 3 project (Marcus et al. 1993), for POS annotation and syntactic annotation and on the TIGER corpus, version 1.0 (Brants et al. 2002) for discontinuous syntactic annotation.

POS error detection results The variation n-gram algorithm for the WSJ found 2495 distinct variation nuclei of n-grams with $6 \leq n \leq 224$, where by distinct we mean that each corpus position is only taken into account for the longest variation n-gram it occurs in. We verified by hand that 2436 of them are errors, which means the method had an error detection precision of 97.6%. Turning from types to tokens, the 2436 variation nuclei that our method correctly flagged as being wrongly tagged correspond to 4417 token annotation errors in the corpus.

Syntactic error detection results For the syntactic annotation in the WSJ, our generalized approach (Dickinson and Meurers 2003b) detected a total of 6277 distinct non-fringe variation nuclei, ranging in length from 1 to 46 tokens. From these 6277, we randomly sampled 100 examples and found that 71 were errors. With this point estimate of .71, we estimate with 95% confidence that the number of errors types detected in the 6277 cases is between 3898 and 5014.

Discontinuous syntactic error detection results Turning to the results of the variation n-gram error detection method for discontinuous syntactic constituents on the TIGER corpus, we obtained a total of 500 shortest non-fringe variation nuclei. Sampling 100 of these, we found that 80 of the 100 samples point to an error. The 95% confidence interval for this point estimate of .80 is (0.7216, 0.8784), so we are 95% confident that the true number of error types is between 361 and 439. The precision thus is comparable to that for continuous syntactic annotation.

3 The proposal

The objective of the project is to explore the prototype error detection method sketched in the previous section as a general method for detecting and correcting annotation errors. For this, the project will examine and extend the variation n-gram method for detecting annotation errors to a) explore its applicability to a wider range of annotation types, b) increase the recall of the method through the refinement of what constitutes comparable contexts, c) add an error correction stage to the error detection approach, and d) research and evaluate the effect of annotation errors and their correction on the use of corpus annotation for human language technology.

a) Error detection for a range of annotation types The method for detecting variation n-grams in principle is independent of the language of the corpus, the nature of the annotation, and the annotation scheme. The method is applicable as long as it is possible to establish a one-to-one mapping between recurring stretches of data and the annotation for which variation is to be detected. To apply the method to different types of annotation one thus needs to make precise which units of data are recurring and how a one-to-one mapping can be established. The project will advance our understanding of the conceptual and technical issues involved in applying variation detection to a broader range of different types of annotation. Given that our previous work has addressed token-based part-of-speech and interval-based syntactic annotation, we next turn to a lexicalized form of structural dependencies, dependency treebanks. The focus on syntactic dependency structures is also motivated by the increasing relevance of such annotation in human language technology and its value as a stepping stone for future work on semantic annotation as another hotbed of current research.

Given the importance of multi-lingual dependency parsing—as evidenced by the upcoming CoNLL-X Shared Task on Multi-Lingual Dependency Parsing—we plan on starting with the Alpino Dependency Treebank of Dutch (van der Beek et al. 2001) and the Prague Dependency Treebank of Czech (Hajič et al. 2001). We envisage that that the definition of variation nuclei we used for discontinuous syntactic annotation can relatively easily be adapted to dependency annotation, so that the central focus of our work will be on determining a relevant notion of context to disambiguate between ambiguity and error.

On the basis of the research for dependency annotation and our earlier work on token-based and structural linguistic annotation, we will investigate how the properties of non-linguistic labeled data, e.g., as used in medical diagnosis validation (Gamberger et al. 1996), relate to those of the explored range of linguistic annotation types.

b) Increased recall through generalizing recurring contexts Our previous work on the variation *n*-gram method shows that the method is relatively precise in detecting errors. But the recall of the method, i.e., the number of errors detected, has received little attention so far. The project will fill this gap by investigating the error detection recall of the method and how it can be increased by generalizing the notion of comparable contexts.

Concretely, we will investigate how to define variation contexts based on more general properties than the words themselves in order to increase recall. Any of the levels of annotation in a corpus, or abstractions thereof, are possible candidates, e.g., the part-of-speech tags of the words in the context instead of the words themselves. This allows nuclei which appear next to low-frequency words to be treated on a par with similar, more frequent constructions. We have started along this path by showing that such a generalization of contexts can double the recall for discontinuous syntactic annotation (Dickinson and Meurers 2005a), but the precision/recall tradeoff needs to be thoroughly explored and investigated with respect to a wider range of annotations. Other context generalizations also seem to be available if one is willing to include language or corpus specific information in computing the contexts. In the WSJ corpus, for example, different numerical amounts, which frequently appear in the same context, could be treated identically. A more general context will be essential for extending our work to machine learning data sets in general. Here a recurring context is an entity outside of the linguistic domain, but it is still based on contextual features which are similar. As a starting point, we will explore connections with the work of Gamberger et al. (1996), who apply a consistency test over data in the medical domain.

In order to expand the number and types of errors we find, it will be beneficial to explore anomaly detection approaches and other statistical techniques for finding unusual data (see, e.g., DuMouchel 1999). Work on fraud detection (e.g., Cortes and Pregibon 2001) can inform linguistic error detection since both avenues of research seek to find data which does not match an underlying model. By exploring the connection between the structure of linguistic and non-linguistic data, we can begin to use techniques from one to help the other. However, Eskin (2000) shows that anomaly detection using linguistically-relevant features only finds 158 errors (with 44% precision) in the same WSJ corpus in which we find 2436. Our research will investigate which role anomaly detection can play as part of a high-coverage error detection method exploiting linguistic properties in the data.

c) From error detection to error correction The project will extend our variation n-gram error detection approach to error correction. To that end, we will develop a taxonomy of the errors provided by the error detection phase, akin to the work in Blaheta (2002). Different kinds of errors require different kinds of correction. Some can be corrected irrespective of the surrounding words, while others are dependent on the context. Still others may need human intervention (cf. Oliva 2001).

Error correction can be cast as a classification problem in which the output is a desired correct tag. In order to determine how labels pattern over the entire corpus, it may be beneficial to apply classifiers such as Naive Bayes, *k-nn*, or decision trees. For the output of the classifier, we can focus our attention only on those spots flagged by the variation n-gram method as potential errors, ignoring all other discrepancies with the benchmark. We then plan to exploit the taxonomy of error types in order to indicate which errors are automatically correctable, which need manual assistance, and which cannot be determined based on the annotation scheme and guidelines. To evaluate the effectiveness of error correction, we will sample the changes to the corpus and evaluate how many were appropriate corrections. As with error detection, for the task of error correction, we will start with POS annotation, with the aim of developing a general method.

Adapting classification methods so that they overcome inconsistent input will likely lead to finding methods which are widely applicable. To correct inconsistencies, one needs to train the classifier to be aware of the kinds of inconsistencies it will see and to learn the general patterns behind them. These inconsistencies stem from ambiguities in the data, which is exactly why classification is difficult in the first place. Thus, by adapting classifiers for the purpose of cleaning the corpus, we will either identify pre-existing classification methods which handle ambiguities the best or design modified methods to handle them better, and we can thereby inform research on classification more generally.

d) Consequences of better corpus annotation The effect of different types of annotation errors and their correction on the use of corpus annotation in human language technology will be investigated by the project. As Padro and Marquez (1998) illustrate, for properly evaluating natural language processing technologies, we need clean testing data. This is true, even if one's technology is robust to training data errors (as, for example, decision trees sometimes are (cf. Mitchell 1997)). Preliminary indications, however, are that erroneous training data is detrimental to learning algorithms, both in the general machine learning literature on class noise (e.g., Zhu et al. 2003; Quinlan 1986) and in the learning of classes in linguistic corpora (e.g., Květŏn and Oliva 2002; Daelemans et al. 1999). Our work will further test the impact of errors on training data by performing the tenfold cross-validation experiment of van Halteren (2000) with automatically corrected training data (manually-checked) against the original data.

It should be noted that the error correction approach proposed in this project is information preserving. With annotation schemes such as XML in wide use, it is easy to incorporate multiple layers of annotation within the same corpus, or to use standoff annotations (e.g., Bird and Liberman 2000). Thus, both the original annotation and the corrected annotation can co-exist in the corpus. Furthermore, although fully automatic correction is desirable, even using suggested corrections as a means of re-annotating the corpus will speed up the corpus work, while ensuring a higher level of accuracy (cf. Brants and Plaehn 2000). Automatic correction software must be used with care, but this is true of any software in use for corpus maintenance.

References Cited

Agrawal, Rakesh and Ramakrishnan Srikant (1994). Fast Algorithms for Mining Association Rules in Large Databases. In Jorge B. Bocca, Matthias Jarke and Carlo Zaniolo (eds.), *VLDB 1994*. Morgan Kaufmann, pp. 487–499.

Bird, Steven and Mark Liberman (2000). A Formal Framework for Linguistic Annotation. *Speech Communication* 33(1-2), 23–60.

Blaheta, Don (2002). Handling noisy training and testing data. In *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing*. pp. 111–116. http://www.cs.brown.edu/~dpb/papers/dpb-emnlp02.html.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius and George Smith (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

Brants, Thorsten and Oliver Plaehn (2000). Interactive Corpus Annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*. Athens, Greece. http://www.coli.uni-sb.de/~thorsten/publications/Brants-Plaehn-LREC00.p%s.gz

Cortes, Corinna and Daryl Pregibon (2001). Signature-Based Methods for Data Streams. *Data Mining and Knowledge Discovery* 5(3), 167–182.

Daelemans, Walter, Antal van den Bosch and Jakub Zavrel (1999). Forgetting Exceptions is Harmful in Language Learning. *Machine Learning* 34, 11–41.

Dickinson, Markus and W. Detmar Meurers (2003a). Detecting Errors in Part-of-Speech Annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, Hungary, pp. 107–114. http://ling.osu.edu/~dm/papers/dickinson-meurers-03.html.

Dickinson, Markus and W. Detmar Meurers (2003b). Detecting Inconsistencies in Treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT-03)*. Växjö, Sweden. http://ling.osu.edu/~dm/papers/dickinson-meurers-tlt03.html.

Dickinson, Markus and W. Detmar Meurers (2005a). Detecting Errors in Discontinuous Structural Annotation. In *Proceedings of the Association for Computational Linguistics (ACL-05)*. Ann Arbor, MI.

Dickinson, Markus and W. Detmar Meurers (2005b). Detecting Annotation Errors in Spoken Language Corpora. In *The Special Session on treebanks for spoken language and discourse*.

DuMouchel, William (1999). Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System (with discussion). *The American Statistician* 53(3), 177–202.

Eskin, Eleazar (2000). Automatic Corpus Correction with Anomaly Detection. In *Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*. Seattle, Washington. http://www.cs.columbia.edu/~eeskin/papers/treebank-anomaly-naacl00.ps.

Fredkin, Edward (1960). Trie Memory. *CACM* 3(9), 490–499.

Gamberger, Dragan, Nada Lavrac and Saso Dzeroski (1996). Noise Elimination in Inductive Concept Learning: A Case Study in Medical Diagnosois. In Setsuo Arikawa and Arun Sharma (eds.), *ALT*. Springer, vol. 1160 of *Lecture Notes in Computer Science*, pp. 199–212.

Hajič, Jan, Barbora Hladká and Petr Pajas (2001). The Prague Dependency Treebank: Annotation Structure and Support. In *IRCS Workshop on Linguistic Databases*.

Hinrichs, Erhard, Julia Bartels, Yasuhiro Kawata, Valia Kordoni and Heike Telljohann (2000). The Tübingen Treebanks for Spoken German, English, and Japanese. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Berlin: Springer, Artificial Intelligence, pp. 552–576.

Květŏn, Pavel and Karel Oliva (2002). Achieving an Almost Correct PoS-Tagged Corpus. In Petr Sojka, Ivan Kopeček and Karel Pala (eds.), *Text, Speech and Dialogue 5th International Conference, TSD 2002, Brno, Czech Republic, September 9-12, 2002*. Heidelberg: Springer, no. 2448 in Lecture Notes in Artificial Intelligence (LNAI), pp. 19–26.

Marcus, M., Beatrice Santorini and M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330. ftp://ftp.cis.upenn.edu/pub/treebank/doc/cl93.ps.gz.

Mitchell, Thomas M. (1997). *Machine Learning*. McGraw-Hill Higher Education.

Nakagawa, Tetsuji and Yuji Matsumoto (2002). Detecting Errors in Corpora Using Support Vector Machines. In *Proceedings of the 17th International Conference on Computational Lingusitics (COLING 2002)*.

Oliva, Karel (2001). The Possibilities of Automatic Detection/Correction of Errors in Tagged Corpora: A Pilot Study on a German Corpus. In Václav Matoušek, Pavel Mautner, Roman Mouček and Karel Taušer (eds.), *Text, Speech and Dialogue. 4th International Conference, TSD 2001, Zelezna Ruda, Czech Republic, September 11-13, 2001, Proceedings*. Springer, vol. 2166 of *Lecture Notes in Computer Science*, pp. 39–46.

Padro, Lluis and Lluis Marquez (1998). On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In *COLING-ACL*. pp. 997–1002.

Quinlan, J. Ross (1986). Induction of Decision Trees. *Machine Learning* 1(1), 81–106.

Skut, Wojciech, Brigitte Krenn, Thorsten Brants and Hans Uszkoreit (1997). An Annotation Scheme for Free Word Order Languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*. Washington, D.C. http://www.coli.uni-sb.de/~thorsten/publications/Skut-ea-ANLP97.ps.gz.

van der Beek, Leonoor, Gosse Bouma, Robert Malouf and Gertjan van Noord (2001). The Alpino Dependency Treebank. In *Computational Linguistics in the Netherlands (CLIN) 2001*, Amsterdam: Rodopi.

van Halteren, Hans (2000). The Detection of Inconsistency in Manually Tagged Text. In Anne Abeillé, Thorsten Brants and Hans Uszkoreit (eds.), *Proceedings of the Second Workshop on Linguistically Interpreted Corpora (LINC-00)*. Luxembourg.

van Halteren, Hans, Walter Daelemans and Jakub Zavrel (2001). Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics* 27(2), 199–229.

Zhu, Xingquan, Xindong Wu and Qijun Chen (2003). Eliminating Class Noise in Large Datasets. In Tom Fawcett and Nina Mishra (eds.), *ICML*. AAAI Press, pp. 920–927.