Unsupervised Detection of Annotation Inconsistencies Using Apriori Algorithm

Václav Novák **Magda Razímová**
Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic
{novak,razimova}@ufal.mff.cuni.cz

Abstract

We present a new method for automated discovery of inconsistencies in a complex manually annotated corpora. The proposed technique is based on Apriori algorithm for mining association rules from datasets. By setting appropriate parameters to the algorithm, we were able to automatically infer highly reliable rules of annotation and subsequently we searched for records for which the inferred rules were violated. We show that the violations found by this simple technique are often caused by an annotation error. We present an evaluation of this technique on a hand-annotated corpus PDT 2.0, present the error analysis and show that in the first 100 detected nodes 20 of them contained an annotation error.

1 Introduction

Complex annotation schemes pose a serious challenge to annotators caused by the number of attributes they are asked to fill. The annotation tool can help them in ensuring that the values of all attributes are from the appropriate domain but the interplay of individual values and their mutual compatibility are at best described in annotation instructions and often implicit. Another source of errors are idiomatic expressions where it is difficult for the annotator to think about the categories of a word which often exists only as a part of the idiom at hand.

In our approach, detection of annotation inconsistencies is an instance of anomaly detection, which is mainly used in the field of intrusion detection. Traditionally, the anomaly detection is based on distances between feature vectors of individual instances. These methods are described in Section 2. Our new method presented in Section 3 uses the data-mining technique Apriori (Borgelt and Kruse, 2002) for inferring high-quality rules, whose violation indicates a possible annotator's mistake or another source of inconsistency. We tested the proposed method on a manually annotated corpus and described both the data and the experimental results in Section 4. We conclude by Section 5.

2 Related Work

Unsupervised anomaly detection has been shown to be viable for intrusion detection (Eskin et al., 2002). The unsupervised techniques rely on feature vectors generated by individual instances and try to find outliers in the vector space. This can be done using clustering (Chimphlee et al., 2005), Principle Component Analysis (Hawkins, 1974), geometric methods (Eskin et al., 2002) and more (Lazarevic et al., 2003).

The difference between our method and previous work lies mainly in the fact that instead using vector space of features, we directly infer annotation rules. The manual annotation is always based on some rules, some of which are contained in the annotation manual but many others are more or less implied. These rules will have their confidence measured in the annotated corpus equal to 1 or at least very close (see Section 3 for definition of confidence). In our approach we learn such rules and detect exceptions to the most credible rules. The rules are learned using the common Apriori algorithm (Borgelt and Kruse, 2002). Previously, rules have been also mined by GUHA algorithm (Hájek and Havránek, 1978), but not in the anomaly detection context.

3 Method Description

Our process of anomaly detection comprises two steps: rules mining and anomaly search.

Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP 2009, pages 138–141,
Suntec, Singapore, 6-7 August 2009.

3.1 Rules Mining

The association rules mining was originally designed for market basket analysis to automatically derive rules such as "if the customer buys a toothpaste and a soap, he is also likely to buy a toothbrush". Every check-out $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is modeled as a draw from an unknown probability distribution Φ, where N is the total number of items available at the store and x_i is the number of items of type i contained in the shopping cart. Further, we define event $E_j = \{\mathbf{x}|x_j > 0\}$, i.e., the event that the shopping cart contains the item j.

In this model, we define a rule $A = (L, R)$ as a tuple where the left side L and the right side R are sets of events E_j. For instance suppose that the toothpaste, toothbrush and soap have indices 1, 2 and 3, respectively. Then the example rule mentioned above can be written as $A_{\text{example}} = (\{E_1, E_3\}, \{E_2\})$, or alternatively $\{E_1, E_3\} \Rightarrow \{E_2\}$. For every rule $A = (L, R)$ we define two important measures: the **support** $s(A)$ and the **confidence** $c(A)$:

$$s\left((L,R)\right) = P\left(\bigcap_{l\in L}(l) \cap \bigcap_{r\in R}(r)\right) \quad (1)$$

$$c\left((L,R)\right) = P\left(\bigcap_{r\in R}(r)\Big|\bigcap_{l\in L}(l)\right) \quad (2)$$

In our example the support is the probability that a cart contains a toothpaste, a toothbrush and a soap. The confidence is the probability that a cart contains a toothbrush given the cart contains both a toothpaste and a soap.

The input of the Apriori algorithm (Borgelt and Kruse, 2002) consists of a sample from the probability distribution Φ, the threshold of the estimated confidence, the threshold of the estimated support and the maximum size of rules. Using this data the Apriori algorithm lists all rules satisfying the required constraints.

In the context of market basket analysis the confidence is rarely anywhere close to one, but in the case of linguistic annotation, there are rules that are always or almost always followed. The confidence of these rules is very close or equal to one. The Apriori algorithm allows us to gather rules that have the confidence close to one and a sufficient support.

3.2 Anomaly Search

After extracting the highly confident rules we select the rules with the highest support and find the annotations where these rules are violated. This provides us with the list of anomalies. The search is linear with the size of the data set and the size of the list of extracted rules.

4 Experiments

4.1 Data and Tools

The experiments were carried out using the *R* statistical analysis software (R Development Core Team, 2006) using the *arules* library (Borgelt and Kruse, 2002). The dataset used was full manually annotated data of Prague Dependency Treebank 2.0 (PDT 2.0). PDT 2.0 data were annotated at three layers, namely *morphological*, *analytical* (shallow dependency syntax) and *tectogrammatical* (deep dependency syntax; (Hajič et al., 2006)). The units of each annotation layer were linked with corresponding units of the preceding layer. The morphological units were linked directly with the original text. The annotation at the tectogrammatical layer was checked automatically for consistency with the annotation instructions (Štěpánek, 2006), however, using our technique, we were still able to automatically find errors. The experimental dataset (full PDT 2.0 data annotated at all three layers) contained 49,431 sentences or 833,195 tokens.

4.2 Experimental Setup and Error Analysis

In our experimental setup, every check-out (i.e., every draw from the probability distribution Φ) contains all attributes of one tectogrammatical node and its governor. The attributes extracted from the nodes are listed in Table 1. Thus every check-out has exactly 52 items, 26 coming from the node in question and 26 coming from its governor.

This being input to the Apriori algorithm, we set the maximal size of rules to 3, minimal support to 0.001 and minimal confidence to 0.995. When the rules were extracted, we sorted them according to the descending confidence and stripped all rules with confidence equal to 1. Using the remaining rules, we searched the corpus for the violations of the rules (starting from the top one) until we found first 100 suspicious nodes. We manually analyzed these 100 positions and found out that 20

Attribute	Description
functor	semantic values of deep-syntactic dependency relations
is_dsp_root	root node of the sub-tree representing direct speech
tfa	contextual boundness
is_generated	element not expressed in the surface form of the sentence
is_member	member of a coordination or an apposition
is_name_of_person	proper name of a person
is_parenthesis	node is part of a parenthesis
is_state	modification with the meaning of a state
sentmod	sentential modality
subfunctor	semantic variation within a particular functor
aspect	aspect of verbs
degcmp	degree of comparison
deontmod	an event is necessary, possible, permitted etc.
dispmod	relation (attitude) of the agent to the event
gender	masculine animate, masculine inanimate, feminine or neuter
indeftype	types of pronouns (indefinite, negative etc.)
iterativeness	multiple/iterated events
negation	a negated or an affirmative form
number	singular or plural
numertype	types of numerals (cardinal, ordinal etc.)
person	reference to the speaker/hearer/something else
politeness	polite form
resultative	event is presented as the resulting state
sempos	semantic part of speech
tense	verbal tense (simultaneous, preceding or subsequent events)
verbmod	verbal mood (indicative, conditional or imperative)

Table 1: Attributes of tectogrammatical nodes used as the input to the rule mining algorithm. Their complex interplay can hardly be fully prescribed in an annotation manual.

of them constitute an annotation error. Examples of extracted rules follow.

$$\begin{array}{ll} & \textit{is\_parenthesis:1} \\ \& & \textit{governor:functor:PAR} \\ \Rightarrow & \textit{governor:is\_parenthesis:1} \end{array} \quad (3)$$

Rule 3 states that if a tectogrammatical node has the attribute *is_parenthesis* set to 1 (i.e., the node is part of a parenthesis) and at the same time the governor of this node in the tectogrammatical tree has its *functor* set to *PAR* (it is the root node of nodes which are parenthesis in a sentence), the governor's *is_parenthesis* attribute is also set to 1. Using this rule we detected 6 nodes in the corpus where the annotator forgot to fill the value 1 in the *is_parenthesis* attribute. There were no false positives and this automatically extracted rule is likely to be added to the consistency checking routines in the future.

$$\begin{array}{ll} & \textit{functor:RSTR} \\ \& & \textit{gender:nr} \\ \Rightarrow & \textit{number:nr} \end{array} \quad (4)$$

Rule 4 states that *RSTR* nodes (mostly attributes of nouns) with *nr* gender (indeterminable gender) also have indeterminable *number*. Our procedure located a node where the annotator correctly determined the *number* as *sg* but failed to recognize the gender (namely, masculine inanimate) of the node.

$$\begin{array}{ll} & \textit{is\_member:1} \\ \& & \textit{dispmod:nil} \\ \Rightarrow & \textit{tense:nil} \end{array} \quad (5)$$

Rule 5, stating that for nodes with *is_member* set to 1 the *nil* value (which means that none of the defined basic values is suitable) of the *dispmod* attribute implicates the *nil* value of the *tense*, is an example of a rule producing false positives.

Due to the data sparsity problem, there are not so many nodes satisfying the premises and in most of them the *nil* value were simply filled in their *tense* attribute. However, there are (rather rare) transgressive verb forms in the corpus for which the correct annotation violates this rule. Many of them were found by this procedure but they are more anomalies in the underlying text rather than anomalies in the annotation. An interesting point to note is that there were several rules exhibiting this behavior with different first premises (e.g., *gender:anim & governor:dispmod:nil ⇒ governor:tense:nil*). The more general rule (*dispmod:nil ⇒ tense:nil*) would not get enough confidence, but by combining it with other unrelated attributes, the procedure was able to find rules with enough confidence, although not very useful ones.

$$\begin{array}{ll} & \textit{resultative:res0} \\ \& & \textit{governor:degcmp:pos} \\ \Rightarrow & \textit{governor:sempos:adj.denot} \end{array} \quad (6)$$

Rule 6 is an example of a successful rule. It states that nodes that govern a non-resultative node and have the positive degree of comparison are always denominating semantic adjectives (i.e., common adjectives such as *black* or *good*). Using this rule we detected a node where the annotators correctly determined the semantic part of speech as *adj.quant.grad* (quantificational semantic adjective) but failed to indicate *degcmp:comp*.

5 Conclusion and Future Work

We have described a fast method for automatic detection of inconsistencies in a hand-annotated corpus using easily available software tools and evaluated it showing that in top 100 suspicious nodes there were an error in 20 cases. This method seem to work best for high-quality annotation where the errors are rare: in our experiments the rules had to achieve at least 99.5% confidence to be included in the search for violations. However, it can also point out inconsistencies in the annotation instructions by revealing the suspicious data points. We have shown the typical rules and errors revealed by our procedure.

The method can be generalized for any manually entered categorical datasets. The rules can take values from multiple data entries (nodes, words, etc.) into account to capture the dependency in the annotation. Other rule-mining techniques such as GUHA (Hájek and Havránek, 1978) can be used instead of Apriori.

Acknowledgement

This work was supported by Czech Academy of Science grants 1ET201120505 and 1ET101120503; by Ministry of Education, Youth and Sports projects LC536 and MSM0021620838.

References

Christian Borgelt and Rudolf Kruse. 2002. Induction of Association Rules: Apriori Implementation. In *Proceedings of 15th Conference on Computational Statistics (Compstat)*, pages 395–400, Heidelberg, Germany. Physica Verlag.

W. Chimphlee, Abdul Hanan Abdullah, Mohd Noor Md Sap, S. Chimphlee, and S. Srinoy. 2005. Unsupervised Clustering methods for Identifying Rare Events in Anomaly Detection. In *Proceedings of the 6th International Enformatika Conference (IEC2005)*, Budapest, Hungary, October 26-28.

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. 2002. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Data Mining for Security Applications*. Kluwer.

Petr Hájek and Tomáš Havránek. 1978. *Mechanizing Hypothesis Formation; Mathematical Foundations for a General Theory*. Springer-Verlag, Berlin, Heidelberg, New York.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. 2006. Prague Dependency Treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia, Pennsylvania.

D. M. Hawkins. 1974. The Detection of Errors in Multivariate Data Using Principal Components. *Journal of the American Statistical Association*, 69(346):340–344.

A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar. 2003. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of SIAM International Conference on Data Mining*.

R Development Core Team, 2006. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Jan Štěpánek. 2006. Post-annotation Checking of Prague Dependency Treebank 2.0 Data. In *Proceedings of the 9th International Conference, TSD 2006*, number 4188 in Lecture Notes in Computer Science, pages 277–284. Springer-Verlag Berlin Heidelberg.