Low Inter-Annotator Agreement = An Ill-Defined Problem?

Vojtěch Kovář, Pavel Rychlý, and Miloš Jakubíček

Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xkovar3,pary,jak}@fi.muni.cz

Abstract. Annotation tasks where the inter-annotator agreement is low are usually considered ill-defined and not worth attention. Such tasks are also considered unsuitable for algorithmic solution and for evaluation of computer programs that aim at solving them. However, there is a lot of problems (not only) in the natural language processing field that are practically defined and do have this nature, and we need computer programs that are able to solve them.
The paper illustrates such problems on particular examples and suggests methodology that will enable training and evaluating tools using data with low inter-annotator agreement.

Keywords: NLP, inter-annotator agreement, low inter-annotator agreement, evaluation

1 Introduction

Inter-annotator agreement (IAA) is considered one of the key indicators of whether a particular classification task is well-defined or not. A lot of attention has been paid to the IAA problem [1,2,3] aiming at not only measuring the agreement, but also excluding the expected amount of agreements by chance, and interpretation of different values of the IAA measurements.

The task is generally considered well-defined, if the inter-annotator agreement is very high, and ill-defined and not worth attention, if the agreement is rather low. To our best knowledge, this is a common view for all classification tasks, through all scientific fields.

In the field of natural language processing (NLP), however, there is a huge number of tasks that do not have naturally high inter-annotator agreement, as people do not agree on the right annotation (even if they are well-educated specialists). Even for such a seemingly straightforward task as morphological tagging (of English), the reported IAA is around 97 percent [4], for more complex tasks like syntactic analysis, information extraction or question answering, it is much much less (e.g. [5]).

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2014, pp. 57–62, 2014. © NLP Consulting 2014

This discrepancy (need for high IAA vs. naturally low IAA in case of most of the NLP tasks) leads to undesirable side-effects. On one side, there are extremely extensive manuals for annotation [6,7] containing hundreds of pages. On the other hand, inter-annotator agreement is rarely published – e.g. the reported IAA for English morphological tagging comes from a semi-official note [4], for Prague Dependency Treebank (PDT [8]), the primary syntactic resource for Czech, there is only one report that describes the annotation of a specific sub-part of PDT [9] which does not report very high numbers in case of important parts of the annotation. It is of course just a speculation, but our opinion is that the results are not published because low IAA numbers would put the whole (mostly very costly) resources in a bad light.

We think both of these effects are really bad, as the aim of all NLP tasks is to learn computers what humans are able to do without any manuals – understand the language – so there should be only minimalistic instructions for any NLP annotation. On the other hand, ambiguity (and low agreement rate) is natural – people often read same sentences differently, and often have to ensure that they understand each other correctly. We can say that low IAA is an integral property of natural languages.

Therefore, we need to be able to handle the tasks with low IAA and use the data with low IAA in training and evaluations, rather than ignore them or try to overcome the fact that the language is ambiguous. This paper suggests a method for using the data with low IAA for meaningful evaluations. We discuss the requirements on such a method, and also some drawbacks and limits of the proposed approach.

2 Problem Examples

In this section we provide real-world examples of the tasks where low IAA causes problems.

2.1 Syntactic Annotation

An example of the project that tried to solve low IAA by extensive manuals for annotation, is syntactic annotation in the Prague Dependency Treebank (PDT [8]), a leading syntactic resource for Czech that we have already mentioned. The manual on the analytical (syntactic) layer has about 300 pages [7], and the annotation procedure was as follows.

Each sentence was annotated by 2 independent annotators, and where they did not agree, there was a third (more experienced) annotator to judge them [8]. So, the one and only syntactic representation available in the treebank is often based on 2/3 biased agreements according to a very complex manual. This procedure is error-prone, and also many of the rules in the manual are debatable. Some of the resulting problems are discussed in [10].

But mainly, the procedure goes against the ambiguous character of the language: In sentences like *"A plane crashed into the field behind the forest"*, it does not matter for correct understanding the sentence, whether the phrase *"behind the forest"* depends on *"crashed"* or *"field"* (although in similar sentences it may be very important). The resulting information is the same. But the annotators need to decide it, and so do the syntactic parsers that are trained and evaluated using this data. This does not correspond to the analysis procedure as it happens when humans analyze the text.

And this is just one example of frequent syntactic ambiguity of many.

2.2 Collocation and Terminology Extraction

Extraction of collocations is an important task for language learners and dictionary makers, to learn or record that in English one says "strong tea" rather than "powerful tea" or "light a fire" instead of "make a fire". However, the agreement among lexicographers on what is good collocation and what is not, is very low [11].

On the other hand, the automatic applications for collocation extraction (e.g. Sketch Engine [12]) are present and they are commercially interesting. They just did not undergo a proper scientific evaluation yet (the procedure reported in [11] is rather debatable, as it uses the same methodology that is used in classification tasks with high IAA), as there is no methodology for evaluating tasks with such a large grey (disagreement) zone.

For extraction of terminology from domain-specific texts, there is nearly the same situation. Terminology (e.g. in form of list of terms) is needed for terminology dictionaries, language learners and consistent translations, and the applications are already there (e.g. [13]). But the agreement on what is and what is not a term in a given domain is very low, and proper evaluation is missing, as there is no methodology available.

The three examples above only illustrate the problem – there are many similar tasks that are neither solved nor evaluated, as they are not "well-defined", however, they are needed and we need a methodology to evaluate them.

3 Methodology Proposal

In this section we present our proposal for annotation and evaluation of tasks with low IAA.

3.1 Requirements

We will illustrate requirements on a binary classification task, which is a typical case (most of the tasks can be straightforwardly reduced to a binary classification task). Let us have classes *Positive* and *Negative*, and the task is to assign each data item to one of these classes. We want to evaluate an automatic tool that does this classification in some imperfect way.

Then, let us have some "gold standard" data annotated by multiple human annotators, that agree in some cases, and disagree in others.

The automatic tool should get positive points for every item assignment into class *Positive*, where all the human annotators agreed that it should be class *Positive*. Similarly for *Negative*. The tool should get negative points for every item assignment into class *Positive*, where all the human annotators agreed that it should be class *Negative*, and vice versa.

We need to be able to handle cases where the annotators do not agree with each other. We propose taking these cases off the evaluation and not count them in at all. Because if even one of the annotators interprets the data differently, the general "human interpretation" is not clear and the automatic tool should get neither positive, nor negative points for any assignment, in these cases.

3.2 Proposed Procedure

Based on the requirements, we introduce a modification of the standard measures *precision* and *recall*, defined for unambiguous gold standard data without human disagreements as follows:

$$precision = \frac{\#true\_positives}{\#true\_positives + \#false\_positives}$$

$$recall = \frac{\#true\_positives}{\#true\_positives + \#false\_negatives}$$

The modified precision and recall will use the same formulas, but with different meaning of *true_positive*, *false_positive* and *false_negative*:

- *#true_positives* will be defined as number of data items where all the human annotations were *Positive* and our tool said *Positive*.
- *#false_positives* will be defined as number of data items where all the annotations were *Negative* but the output of our tool said *Positive*.
- *#false_negatives* will be defined as number of data items where all the annotations were *Positive* and our tool said *Negative*.

In other words, we firstly remove all the data items where the human annotators disagree, and then measure standard precision and recall on the rest.

This idea can be easily generalized to classification into more classes. In that case, however, we may want to give some positive points in addition if the output of our tool agreed *at least with one of the annotators*, and/or negative points if the output of our tool agreed *with none of the annotators, even if they did not agree*. In this case, we will be able to somehow use the data even if the annotators disagree. However, this approach brings more complexity into the evaluation and it may me better to transform the problem into a binary classification task, which is possible in most cases, and transparent.

4 Discussion

There are some difficulties with the above introduced procedure. In the following points we mention them and discuss possible solutions:

- On the first sight, the procedure increases the price of the testing data, as many of the annotations (all cases where the annotators disagreed) are not used. However, the data will record ambiguity and will be of higher quality than if we attempt to *decide* the disagreements. Therefore, also the evaluations will be more sound, and automated learning from such data will be able to be more informed.
- We need to count with random agreements, especially when the part of the data where people disagree is rather big. Probability of random agreements can be easily computed for most of the classification tasks, and can be trivially decreased by increasing the number of annotators. For example, if probability of *Positive* judgement is 50%, increasing number of annotators to 7 will reduce the number of random agreements below 1%. Of course, sometimes it will mean increasing costs again. On the other hand, in most of the tasks the probability of the *Positive* judgement is much lower.
- In cases where the agreement is really low, the question whether the task is well-defined or not, will persist. The maintainers of the data should check carefully if the level of disagreement corresponds to the real ambiguity of the task and correct the annotation instructions if not. It is not easy to introduce a quantitative algorithmic test here, as the level of ambiguity significantly varies among various tasks.

5 Conclusions

We have introduced a new view on classification problems where people often disagree, mainly from the perspective of natural language processing, but suitable for any other field. We have proposed a methodology for using data with disagreements for testing (and partly training) of automatic classification tools. The proposed method is straightforward and easily applicable to any data.

We believe that in the future the data with disagreements will not be considered radioactive and they will be used for serious research. Also, we believe that the idea will encourage data maintainers to publish their agreement figures consistently.

Acknowledgements This work has been partly supported by the Ministry of Education of the Czech Republic within the LINDAT-Clarin project LM2010013 and by the Czech-Norwegian Research Programme within the HaBiT Project 7F14047.

References

1. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1) (1960) 37–46
2. Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational linguistics **22**(2) (1996) 249–254
3. Fleiss, J.L., Levin, B., Paik, M.C.: Statistical methods for rates and proportions. John Wiley & Sons (2013)
4. Manning, C.D.: Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In: Computational Linguistics and Intelligent Text Processing - 12th International Conference, CICLing 2011. Springer, Berlin (2011) 171–189
5. Sampson, G., Babarczy, A.: Definitional and human constraints on structural annotation of English. Natural Language Engineering **14**(4) (2008) 471–494
6. Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M.A., Schasberger, B.: Bracketing guidelines for treebank II style Penn treebank project (1995) `languagelog.ldc.upenn.edu/myl/PennTreebank1995.pdf`.
7. Hajič, J., Panevová, J., Buráňová, E., Urešová, Z., Bémová, A., Štěpánek, J., Pajas, P., Kárník, J.: Annotations at analytical level: Instructions for annotators (2005) `ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/pdf/a-man-en.pdf`.
8. Hajič, J.: Complex corpus annotation: The Prague dependency treebank. Insight into the Slovak and Czech Corpus Linguistics (2006) 54
9. Mikulová, M., Štěpánek, J.: Annotation procedure in building the Prague Czech-English dependency treebank. In: Slovko 2009, NLP, Corpus Linguistics, Corpus Based Grammar Research, Bratislava, Slovakia, Slovenská akadémia vied (2009) 241–248
10. Kovář, V., Jakubíček, M.: Prague dependency treebank annotation errors: A preliminary analysis. In: Proceedings of Third Workshop on Recent Advances in Slavonic Natural Language Processing, Brno, Masaryk University (2009) 101–108
11. Kilgarriff, A., Rychlý, P., Jakubíček, M., Kovář, V., Baisa, V., Kocincová, L.: Extrinsic corpus evaluation with a collocation dictionary task. In N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (2014) 1–8
12. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: Ten years on. Lexicography **1** (2014)
13. Kilgarriff, A., Jakubíček, M., Kovář, V., Rychlý, P., Suchomel, V.: Finding terms in corpora for many languages with the Sketch Engine. In: Proceedings of the Demonstrations at the 14th Conferencethe European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, The Association for Computational Linguistics (2014) 53–56