Case study of BushBank concept

Marek Grác

Faculty of Informatics
Masaryk University
Czech Republic

Abstract. In this paper, we present a new type of annotated corpus, called BushBank, which improves handling of ambiguity in natural language. Unlike in traditional approaches where data are directly disambiguated, in a BushBank, disambiguation is done later, based on application needs. This has major impact on the structures used in the corpus, since ordinary syntactic trees disallow ambiguity. Our approach was tested on 10.000 sentences and more than a hundred annotators when creating Czech BushBank. The paper contains information about creating such a resource and the methods used to obtain high inter-annotator agreement.

Processing natural language is one of those areas where the quality and the quantity of lexical resources distinguish a great project from an inferior one. Lexical resources for major languages tend to address both these requirements, but the situation for smaller languages varies a lot. Thanks to various projects like WebBootCat [1], it is possible to build large corpora from documents available on the internet. Unfortunately this approach can't help us in the process of building high-quality annotated corpora. The most noticeable examples of annotated corpora are the PENN Treebank [11] and PDT [7]. Sentences in these corpora are parsed, i.e. annotated with (at least) syntactic structures. These structures are offered to users in the form of syntactic trees and they are unambiguous. In order to obtain high quality, unambiguous annotation of natural language (which is ambiguous on every level), skilled annotators and hundreds of pages of manuals [8] are needed. Building such complex lexical resources is out of reach for most of the less-used languages.

This paper focuses on a new type of annotated corpus named BushBank and an example study performed on Czech language that belongs to the Slavic languages together with Russian or Polish. Slavic languages tend to very good for such experiments as they have rich morphology and fairly free word order. Czech language is one of the most described European languages and there are already several high quality resources like Prague Dependency Treebank [7] or a Czech version of EuroWordNet [13]. This gives us a reference to compare our results to.

25th Pacific Asia Conference on Language, Information and Computation, pages 353–361

1 General overview of BushBank

Language resource, from our experience, tend to focus on complex and clear description of a language. Such description is extremely useful for language research and theory observation, but its usage in NLP applications is limited. Bushbank (derived from TreeBank) was created to fulfill those needs and is almost purely application-driven. We are aware that different applications have different needs and thus several levels of annotation are needed. Various levels of annotation are already used in other language resources. The main difference is that ambiguity in annotations is not resolved on the corpus level, but we offer tools to obtain *correct* annotations based on different needs for the quality/quantity ratio of the data.

1.1 Annotation principles

Annotation of linguistic data is considered to be an expert task. This is especially true for those corpora that attempt to cover several layers of a language. Extensive training for annotators is required, together with an exhaustive annotation manual (annotation layer for syntactic level of PDT2.0 spans 301 pages). Our funding situation and the lack of trained annotators did not allow us to use any of the existing heavy-weight approaches.

We had to find a more light-weight approach that will be affordable even for smaller budget and less-widely-used languages. Inspired by existing projects, we decided to replace experts by untrained annotators. These techniques, called crowdsourcing, are widely used on the internet (e.g. Wikipedia) and we have observed use of these approaches for partial linguistic annotation [12]. However we are not aware of any attempts to build annotated corpora entirely via crowdsourcing.

In order to be able to use crowdsourcing we need access to a crowd that exceeds critical mass. This may be a problem even for large languages, but for these it is possible to use paid services, lika Amazon Mechanical Turk. For smaller languages (e.g. Czech - 10 million speakers), we are unable to target a suitable crowd even when using such paid services, since these simply do not exist. These constraints lead us to the decision to target students who were taking courses related to NLP. Our annotators were mostly in their first year at the university, they have very limited amount of deeper linguistic knowledge and no previous experience with annotation. We believed that their interest in the topic of NLP should provide better results compared to a *general* crowd. Students should complete their projects in 10-20 hours, meaning that extensive manuals had to replaced by much simpler rules, which can't cover less frequent or borderline cases in any way other than the intuitive one.

We assume that an annotation standard is usually an attempt to approximate several mutually exclusive and contradictory constrains [9]:

1. **completness**: the annotation should provide complete linguistic insight into the particular area

2. **consistency**: the annoation should be consistent, i.e. same or similar language phenomena should be handled in same or similar ways
3. **usability**: the annotation should enable straightforward usage in the intended application
4. **simplicity**: the annotation should be as simple as possible to make high inter-annotator agreement achievable.

In our experience, most language resources try to find a trade-off among the constrains by prioritizing them in order given above. They prefer completness over consistency and both of them over simplicity. Following the so-called KISS principle, we are strongly convinced that the reverse order of those constrains represents a much better priority list. Thus our priorities are:

1. **simplicity**: so that annotators do not err too often and can rely on their introspection for simple cases
2. **usability**: annotated data have to be easily applied to other NLP applications
3. **consistency**: if annotators do not err and rules are simple, consistency follows
4. **completness**: it will be nice but we do not rely on it

Main objection against this new order of priorities can be that consistency is crucial in most NLP applications. This applies to using the data both for testing/development and for machine learning. From our perspective, natural language is too ambiguous and flexible to be easily and consistenly annotated. In real world, even with trained annotators and exhaustive annotation manuals, we have to face situations where annotators encounter the possibility of using more than one correct annotation. We do not attempt to force selection of one preferred annotation, but face the fact that preferred solution is intuitively different per annotator. Inconsistencies between annotators are traditionally resolved by an expert who decides which annotation is correct [14]. Such qualified opinion improves consistency of annotations. Our approach to building corpora does not believe in experts and border-line cases have to be selected by the annotator according to their introspection and intuition. Inconsistencies between annotators are not handled directly in the corpus, but in an external framework where the user, on a per-application basis, selects the proper resolution model that gives him the desired results. For some applications, it is optimal to use all results that are confirmed by 5 out of 7 annotators, but for other purposes we want 100% agreement. We offer all annotations with the corpus, which makes it easy to create a new resolution model based on our unique requirements. The fact that we do not offer ultimate correct annotation leads to an impossibility of *completness* for non-trivial cases. To obtain completness we will have to offer annotators superset of all valid answers what is not possible with automatic tools and obtain perfect inter-annotator agreement. Therefore, the last position that completeness occupies on the priority list is in fact a necessity.

Simplicity as a top priority is, on the other hand, necessitated by the fact that our annotators do not have enough time for training and that they will annotate

for only a few hours. Their ability to understand their task is crucial to success of our project. In order to obtain high quality and high speed of annotation, we have decided to constrain the annotators as much as possible, even if completness will be hurt (we expect that to happen anyway). Annotators can only select the correct answer from a list, which is generated by existing automated NLP tools. This means that there is no way for the annotator to add a correct answer if it is missing. In our existing work, the annotators had to choose among two (valid/invalid noun phrase), or among up several answers (select verb to which the noun phrase relates). Limiting creativity and working with pre-processed data helps us to speed up the process, to increase inter-annotator agreement and (therefore) also to increase consistency.

1.2 Annotation manual

The lack of time for exhaustive training of annotators has to be supplemented by an annotation manual. We have developed and tested manuals for two tasks. The first one is the identification of syntactic elements (short noun phrases, coordinations, verb phrases and clauses) and the second is creating dependencies between noun phrases and their parent elements. In our case study, manual for identification of syntactic elements was at first based on just first two hundred sentences and it contained 21 rules and 13 examples. It is clear that a manual of this size does not even try to describe what a noun phrase is. Instead, it just describes some of the most common borderline cases where annotators are not sure about the correct answer. After a year of development and more than a hundred annotators, we ended up with a manual consisting of 48 rules (most of them are defined by a single clause) and 70 examples.

The second annotation manual should help annotators to find dependencies between identified valid noun phrases and their parents elements. This task is much more ambiguous than the previous one but we have found that most of the cases can be solved by annotator introspection and we do not have to define exact rules for them. First version of the manual contained just 6 rules and the current one contains 8. In our view, examples are at least as important as the rules themselves.

1.3 Corpus structure

BushBank is a concept that extends TreeBanks, which are sets of annotated syntactic trees, by reducing the requirements for unambiguity and making them closer to real language. Like other modern corpora, bushbank usually covers several layers of lingusitic annotation. For this reason, we have decided to use NXT NITE [3], which was developed for multimodal corpora. We do not plan to have a multimodal corpus, but using existing libraries for complex search queries and the XML format persuaded us. On top of this toolkit, we have built our own library which maps elements in the corpus to objects, so that programmers do not need to care about the internal NXT NITE structures or about XML elements.

One of our main objections against existing annotated corpora is the fact that they treat language as an unambigious structure and possible ambiguities are solved by the annotation manual or by expert decision. This leads to a situation when corpora users are not able to determine whether they are handling cases that were easy to determine or cases where even human annotators were not really sure. For various NLP applications, it is crucial to know whether the application can handle correctly at least the clear cases and only later focus on areas which are hard even for humans.

Ambiguity in a bushbank is one of its main advantages. In fact, only the first layer has to be disambiguated. This layer contains marks for sentences and a token for every word in the corpus. We are aware that even on this layer, it is possible to have ambiguities but both simplicity and usability will be corrupted if we introduce ambiguity at this level. Currently for the Czech BushBank (as first case-study of bushbank concept) we have the following layers:

1. **tokens**: contains tokens and marks for begin/end of sentence.
2. **morphology**: defines lemma and morphologic tag for tokens.
3. **syntactic structures**: defines short noun phrases, verb phrases, coordination and clauses. This structures uses the token layer.
4. **relations between syntactic structures**: for every short noun phrase we define its dependency parent.

We believe that corpus users should be able to select proper resolution model for their needs and thus they should have access to the existing annotation also in the form of raw data. All our results are easily reproducible and can be reproduced by anyone interested in doing so. To showcase the importance of access to raw data and deducing *correct answer* on the application level, we would like to present two existing applications. The first one uses the Czech BushBank as a test suite for the rule based syntactic analyser SET [10], where we want to determine its success only on those syntactic structures and their relations which are identified correctly by all (number vary from 3 to 7) annotators, as the cases where human annotators differ in opinion are the difficult cases and can be handled after the simpler part is done. The other is the project Shallow Ontology based on Valency Frames [6], where semantic network is created manually from candidates obtained by mapping noun phrases to possible semantic roles. In this case, we are interested in the semantic head of a noun phrase and as the annotator has no access to the context of the noun phrase, we do not care whether the noun phrase is a longest correct one or not (eg. the blue little rabbit and little rabbit belongs to same semantic group).

2 Obtaining high inter-annotator agreement

One of the main features of an annotated corpus is the quality and thus reliability of its data. In the previous section, we have shown that it is possible to choose a tradeoff between quality and quantity on an application level. However, we still plan to obtain data as good as possible (given the constraints of time, experience

and training). To prove our theory in practice, we have created a case study. In this study, we have built the Czech BushBank, which contains 10.000 sentences annotated by more than one hundred annotators and where each sentence is annotated at least three times. The experiment was divided into several stages and we tried to make the conditions the same for every person during any given stage. Experience from previous stages was incorporated into later ones, usually in form of an upgraded annotation manual.

Using unskilled annotators makes interpretation of the results even more difficult than when working with classical models. We really can't (and don't want) to say which answer is correct but we wish to know that this answer is really preferred by annotators and it is not just a random coincidence. This can be a major problem when dealing with identification of syntactic elements as annotator answers are just yes/no. In our case, if we expect to have 80% of the pre-processed data correct, then if two annotators will select answers randomly (and ratio of valid/invalid answers remains) then they can obtain $0.8 * 0.8 + 0.2 * 0.2 = 64.04\%$ agreement. These reasons lead us to use a more standard solution for computing the inter-annotator agreement.

Problems of agreement by chance are especially pertinent when we are using Cohen's kappa [4] to measure inter-annotator agreement. This coefficient takes into account the number of possible answers and frequency of their usage. Cohen's kappa can be applied to evaluate agreement between two or more annotators. For illustration purposes, we have decided to compute it only for pairs of annotators as these are the numbers that are usually published and can therefore be compared readily. Interpreting the obtained numbers can be difficult for those who are unfamiliar with Cohen's kappa. For this reason, several benchmark scales were developed (table 1 presents most used metrics).

Table 1. Example of Benchmark Scales for Cohen's Kappa

Landis and Koch		Fleiss	
Kappa Statistics	Strength of Agreement	Kappa Statistics	Strength of Agreement
< 0.0	Poor		
0.0 to 0.20	Slight		
0.21 to 0.40	Fair	< 0.40	Poor
0.41 to 0.61	Moderate		
0.61 to 0.80	Substantial	0.40 to 0.75	Intermediate to Good
0.81 to 1.00	Almost Perfect	0.76 to 1.00	Excellent

Since we have no gold standard, we have to compare the quality of annotation just between the annotators themselves. If the inter-annotator agreement is high, we can assume that the annotators are able to work consistently and according to the annotation manual and their linguistic knowledge. If agreement is low then the whole data set is unreliable and our resolutions model won't help us.

The threshold of acceptability varies among authors, but the general consensus seems to be that Cohen's kappa over 0.67 is reliable enough ([2], [5].

For the sake of simplicity, we will focus just on identification of syntactic elements in the first and the last stage of our project. In the first stage, the annotators had a simple annotation manual based on annotations of two hundred sentences. All of these sentences were annotated with a rule-based syntactic analyser, making the ratio between valid and invalid elements fairly high. Cohen's kappa obtained in this stage was on average on the border of acceptability. In table 2 we can see the worst, one of the average and the best results of annotation. In the worst case the annotator did not understand the concept of verb phrases and there were $3 - 5$ times as many errors as usually. Although these data are not reliable enough when used directly adding a third annotator helps us to make these results reliable enough for practical application. But we do need 50% more annotators and the price of an annotated sentence has raised considerably.

Table 2. Annotatition results 1st stage

valid by annotator A/B	invalid by annotator A/B	agreement	cohen kappa	max kappa
2377/2604	975/793	79.04%	0.50	0.85
3009/2669	326/685	88.43%	0.66	0.81
2473/2431	678/713	92.68%	0.76	0.96

In the last stage of our case study, annotators had access to an annotation manual which was more detailed and mainly consisted of more examples and also the graphical interface was optimized to suit their needs better. The rules in the manual were just described in more detail and in the case of *easy* to identify elements there were no changes relative to previous annotations. Additionaly an output from three more syntactic grammars was added, so they had to identify more elements and ratio between valid/invalid has lowered, since many of the correct elements were found multiple times but the errors produced by different grammars were mostly disjoint.

Results of the last stage are presented in table 3. In this case, we did not detect any noticeably bad or great annotations. For that roason, we have decided to show a table containing three annotators working on the same data set. The consistency between those is on high level and easily exceed an acceptability treshold of 0.67.

3 Conclusions

In this paper, we have presented a new type of corpus, BushBank, that attempts to cover ambiguity in a natural language more precisely. We have described the

Table 3. Annotatition results 3rd stage

annotators	agreement	cohen kappa	max kappa
D vs M	0.91	0.77	0.92
D vs P	0.94	0.85	0.99
M vs P	0.90	0.77	0.92

basic ideas and the structure of such corpora, together with our experience on applying this theory. As a case study, we have presented Czech BushBank that contains 10.000 sentences and which was built entirely through the presented techniques. We have shown that even with a small annotation guide and annotators without proper training, it is possible to obtain high inter-annotator agreement and create high-quality data.

The infrastructure for building this type of annotation corpus is freely available at `http://www.bushbank.org` and we will gladly help with creation of a similar resource for other languages. We believe that this can help even smaller languages in obtaining valuable linguistic resources, using a very low-cost approach.

Acknowledgments

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and by the Czech Science Foundation under the project P401/10/0792.

References

1. M. Baroni, A. Kilgarriff, J. Pomikálek, and P. Rychlỳ. Webbootcat: instant domain-specific corpora to support human translators. In *Proceedings of EAMT*, pages 247–252. Citeseer, 2006.
2. J. Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, 22(2):249–254, 1996.
3. J. Carletta, S. Evert, U. Heid, and J. Kilgour. The nite xml toolkit: data model and query language. *Language resources and evaluation*, 39(4):313–334, 2005.
4. Jacob Cohen. A coefficient of agreement for nominal scales. In *Educational and Psychological Measurement*, volume 20, pages 37–46, 1960.
5. B.D. Eugenio and M. Glass. The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101, 2004.
6. M. Grác. Shallow ontology based on verbalex. In *Slovko 2009*, page 114, 2009.
7. J. Hajič. Complex Corpus Annotation: The Prague Dependency Treebank. Bratislava, Slovakia, 2004. Jazykovedný ústav Ľ. Štúra, SAV.
8. J. Hajič, J. Panevová, E. Buráňová, Z. Urešová, J. Štěpánek, P. Pajas, and J. Kárník. Anotace na analytické rovině – Návod pro anotátory, 2005.

9. Miloš Jakubíček, Vojta Kovář, and Marek Grác. Through low-cost annotation to reliable parsing evaluation. In *PACLIC 24 Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, 2010.
10. V. Kovář, A. Horák, and M. Jakubíček. Syntactic analysis as pattern matching: The SET parsing system. In *Proceedings of the 4th Language & Technology Conference*, pages 100–104, Poznaň, Poland, 2009.
11. M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.
12. Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily. Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 122–130, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
13. K. Pala and P. Smrž. Building Czech WordNet. *Romanian Journal of Information Science and Technology*, 7(2-3):79–88, 2004.
14. B. Snyder and M. Palmer. The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, 2004.