UNCORRECTED DRAFT. For the final version, see Automated Building of Error Corpora of Polish, in: B. Lewandowska-Tomaszczyk (ed.), *Corpus Linguistics, Computer Tools, and Applications – State of the Art. PALC 2007*, Peter Lang. Internationaler Verlag der Wissenschaften 2008, s. 631-639.

Marcin Miłkowski
Institute of Philosophy and Sociology
Polish Academy of Sciences

AUTOMATED BUILDING OF ERROR CORPORA OF POLISH

1. Introduction

For most languages, including Polish, big error corpora are lacking. Traditional error corpora are collected and annotated by linguists, and the process is manual or only slightly automated. The task is therefore tedious and costly, and the results represent linguists' knowledge about correct usage. This requires additional work to avoid theory-laden distortion of data.
In this paper, I will show how to automatically develop error corpora by using revision histories of documents. The idea is based on a hypothesis that most frequent minor edits in documents represent corrections of typos, slips of the tongue, grammar, usage and style mistakes. This hypothesis has been confirmed by frequency analysis of the revision history of articles in the Polish Wikipedia. Partial results of the analysis and perspectives for integrating the error corpus with the Polish National Corpus will be presented.

2. Gathering error corpora

The traditional way to prepare an error corpus is to manually annotate it with extended information on grammar mistakes, style abuses, misspellings, typos, etc. The manual annotation requires that the corpus be prepared by a qualified linguist or a language professional (a proof-reader in a publishing house, a reviewer in a translation agency etc.) This however means that collecting large error corpus will be time and resource costly.
There were some small corpora collected this way for Polish (Pisarek 1978 is a dictionary based on sampling newspaper texts). In case of creating larger corpora, the costs could turn out prohibitive. As one of the uses of the Polish error corpus is to build an open-source Polish style and grammar checker rules for LanguageTool correction software, which is a non-profit project, scarcity of resources is one of the main factors at play.
One of the ways to reduce the amount of work required is to reuse already existing resources. For example, one could gather the results of newspaper proofreaders work (Sågvall Hein 1998). In some environments, proofread texts are gathered systematically. For example, large translation agencies try to ensure constant quality assessment, and introduce systematic Language Quality Inspection (LQI). The problem with reusing the data generated this way is that they are usually considered confidential, which is also what clients of translation bureaus require. The relevance of such data for linguistic research, and especially for research on common grammar mistakes, etc., is clear. Yet building the corpus depends in such a case on the eagerness of commercial institutions to contribute their material to the corpus. One may also expect is that the error corpus gathered this way cannot be representative for an average language speaker; it reflects linguistic competence of highly skilled and usually specially trained professionals. Whether this is a benefit or a flaw, depends on the projected application of the corpus.
Another way of building an error corpus is to gather errors that are frequent in language examinations, whether specially organized, or being a part of the normal school curriculum (such a corpus can also contain learners of a language as a second one). School error corpora will reflect

lower linguistic competence but such data could be useful for refining methods of spelling, style or grammar teaching. However, the amount of work that the standard examination centers are required to do is so immense that one cannot expect that they will be willing to help create even a simple statistic of errors. In other words, such a process needs special funding. It could be however very useful to organize national-level tests for a balanced sample of pupils, as it is done for the PISA/OECD research (Adams and Wu 2002). The benefit of this procedure is that annotation can be based on standard school error terminology, and would probably not require any training for teachers.

Both kinds of corpora – containing language professionals' usage and language learners' usage – can hardly be seen as representative. The first group is supposed to be better skilled, and the second will always show lower competence. For a general research on the language use, a less biased selection of language users is desirable. In order to be able to use statistic methods on the corpus, it is required that the corpus is big enough. In case of really big corpora, manual annotation could seem infeasible. Automatic methods must be used.

Using an existing software grammar checker, one could annotate any corpus. However, the errors annotated will directly depend on the quality of the checker, and on the scope of errors it is able to detect. As for Polish, no representative grammar checker exists now; such a scenario is at best useful in the future. It is however quite useless for creating an error corpus that will be used exactly for research on which errors should be detected by a planned grammar checker. In principle, one could try to use machine-learning techniques to process an already error-annotated corpus and see whether using some algorithms (for example, some rule-learning taggers could be used), the results could be generalized and extrapolated. But it should come as a surprise if an extrapolation of a couple of existing rules would result in an representative set of detected errors. It is also clear that the original grammar checker would introduce theoretical bias into the corpus.

That is the reason why it seemed essential to create an unbiased data set. The hypothesis used for it was that resources such as revision history in Wikipedia, Wikia, and other collaborative editing systems, could be turned into corpora of errors, just by extracting the minor edits.

The most theoretically interesting aspect is that the corrections will represent the average speaker's intuitions about usage, and this seems to be a promising way of researching normativity in claims about proper or improper Polish. The method can be however used also for processing the edits done by language professionals or teachers. For example, proof-editors rarely annotate or comment their corrections; they simply correct errors. Analyzing the patterns of corrections made can provide insight into the *implicit* linguistic knowledge they are using but might be unable to report verbally. The same is true for processing examination results; all that is required is that they are available electronically with corrections (or at least with some kind of error markup). Probably in the future, some examinations centers will use more and more automatic error marking (especially for multiple-choice tests). The data gathered this way can be also processed using the method presented here; however, it should be understood that different language user groups might result in different distribution of errors and mistakes annotated.

3. Processing the revision history

By processing the revision history, we can gain pairs of segments in the corpus: first representing the error, and the other representing the correction. Moreover, it is relatively easy to tag parts of speech, compare subsequent versions, and prepare a text file containing the resulting corpus. In other words, the revision history – the original document and its subsequent versions – can be turned into a set of changes, or edits, such as inserting the word, deleting the word, or changing the word order (which is a combination of both operations).

In many environments, revision histories are automatically generated. For example, control version systems are used in large legal firms, official agencies; shortly: in collaborative work environments, where it is crucial to be able to assign responsibility for a change. On the web, control version modules are built into wiki editing systems. This enables us to reuse easily all the information

contained in the revision history. (Such control version systems can be introduced into the work flow of a newspaper in order to build an error corpus without any proof-editor intervention.)
Some systems do not have any control version systems in place. If they present frequently changed and corrected content on the web (a newspaper website seems to be an obvious example), the website changes can be monitored without installing any software on the editing side. Publicly available documents which are corrected frequently and are not copyrighted would be an excellent target of website monitoring.
The generated set of changes can be processed statistically. The most important factor is frequency; yet the structure of the error – the pattern of its appearance – could also be statistically discovered (by using standard collocation measures, for example). The bottom line is that frequent and minor corrections, which are not simple formatting or other conventional changes, most probably represent frequent errors. Larger corrections tend to be substantial, i.e., they are either deletions of information or additions of new information. The results reviewed below hint that this hypothesis is true. Though the initial data sample seemed large, it is still insufficient to establish statistically robust inferences that would confirm the hypothesis beyond a reasonable doubt.
The important task is to filter out formatting changes to leave only the significant but still minor corrections. Roughly, an addition/deletion of less than five words could be considered a minor change; but until we have analyzed enough data, the exact figure is not easy to determine, and it could be highly dependent on the grammatical features of the language being analyzed. The easiest and safest rule of the thumb is to allow all corrections into the corpus. It will be queries to the corpus search engine that will select minor edits.

4. Method evaluation

The approach was evaluated practically using a Polish Wikipedia history file downloaded at the beginning of 2007 (about 30GB large XML file, about 2440 millions of words, about 17GB in pure text). Wikipedia as such cannot be a source of representative information about an average language speaker because of such factors as Digital Divide (poor people tend to use computers less frequently), higher education levels, and uncommon theme scope. For example, historical data about churches is not so common in everyday speech as in encyclopedias.
However, Wikipedia authors tend to make specific mistakes that a grammar checker should detect: even when they check the spelling, they do it for single words, and not for chunks (not contextually) – as most computer spell-checkers ignore all context information. This way they are prone to specific computer typos: writing a single word as two words ("w raz" instead of "wraz", or "a pro po" instead of "à propos".) This kind of errors is especially important for planning a rule set for a grammar checker.
Higher education level of Wikipedia users (see http://pl.wikipedia.org/wiki/Wikipedia:Wikipedy%C5%9Bci_wed%C5%82ug_wykszta%C5%82cenia) can also be used to justify the claim that their language use embodies a linguistic norm; it is usually assumed that educated writers are the source of normative decisions in the linguistics (see for example Klemensiewicz 1966, p. 17).
The Wikipedia history file is not an explicit record of changes. It contains simply "snapshots" of all subsequent article versions. So extracting only the changes requires additional processing. For evaluation purposes, the general Unix program *diff* was used for processing.
Producing the corpus involved the following steps:

1. Filtering out formatting elements
2. Sentence-level tokenization
3. Word-level tokenization
4. Comparing subsequent versions.
5. Frequency analysis of various minor edits

Step 1. Filtering. Wikipedia contains special formatting markup that is often corrected (so-called "wikization" of an article). Otherwise, a large number of non-relevant changes would be included in

the corpus. Robots, called "bots," which automatically adjust date formats or make other conventional changes, commit lots of changes.
Step 2. Sentence-level tokenization. Although this step is optional, and was not made in first experiments, it is useful to be able to refer to the sentence start and sentence end when trying to define the pattern of the error.
Step 3. Word-level tokenization. This step is strictly necessary. Word boundary is defined syntactically, as whitespace or punctuation sign. It is up to the researcher to decide whether it is useful to include hyphens on the list of punctuation signs.
Step 4. Comparing versions. This is the most time consuming step in the process. The *diff* program was instrumented using simple AWK scripts. As a result, a standard *diff* file (in unified diff format) was created. The file is about 3GB large.
Step 5. Frequency analysis. Simple AWK scripts on the output file generated required results. Though the input corpus file seems large, the simple test for frequent changes of single words resulted in a 11MB file; in case of frequent changes of two or more words the file was only a little larger (14MB). Moreover, most edits are less frequent than 1000, which makes statistical analysis harder. In other words, Polish Wikipedia history file is not yet large enough to provide a lot of new meaningful information. (The detailed analysis of frequent errors in Polish Wikipedia is out of the scope of this paper).
The top “single word change” query results are presented below:

#	Original	Correction	Frequency
1	województwie	województwa	26162
2		–	16003
3	zamieszkiwało	zamieszkiwały	2738
4	zamieszkiwały	zamieszkiwało	2646
5	podstawowe	przegląd	2519
6	sie	się	2343
7			1854
8	,	,	1760
9	też:	też	1753
10	(†	(zm.	895
11	Gmina=Saint	Gmina=Saint	851
12	także	też	834
13		w	797
14	–		762
15	to		587
16	to	–	584
17	E.	Edward	512

Table 1: Most frequent single word changes in Polish Wikipedia.

The first five entries are connected with a conventional change (executed by a bot – it is impossible to filter out bots in *a priori* fashion*)* in many geographical articles. This means that in order to keep the corpus clean, these conventional changes should be filtered out. The easiest way is to review the frequent corrections and subsequently add the required filtering code to the existing wiki markup

filtering scripts. The sixth change (from “sie” to “się”, i.e., adding a Polish diacritical mark in a popular typo) is a bit of valuable information.
Manual analysis of the results helped write new Autocorrection rules for OpenOffice.org (the new rules are scheduled for 2.3 release of the package); the previously existing rules were based on a random variation of misspellings. Those randomly generated data does not seem appear frequently in Polish corpora (including Polish Wikipedia revision corpus) nor on the Polish web. It seems therefore they were artifacts of a badly designed algorithm.
After ignoring the conventional changes, the original hypothesis seems confirmed: most frequent errors could be added to the Autocorrection file as they represented most frequent mistakes and misspellings, yet there were some frequent changes that resulted apparently from style changes executed by bots. The stylistic changes may but do not have to be introduced only when there were errors. A perfectly grammatically correct sentence can often be rewritten to sound better in the context (e.g., change #12 is clearly stylistic: it is a replacement with a synonymous word).
Unfortunately, only some style changes can be detected automatically. In Polish style manuals, it is often advised not to use the same word in the same paragraph more than once (in reality, the rule applies only to some words but the rule is never formulated explicitly with caveats and exceptions). Accordingly, users replace the repeated (or otherwise undesirable) words with synonyms. Using a synonyms dictionary, it could be possible to filter out such changes automatically or to analyze them deeper. Nevertheless, the wording changes etc. could be harder to detect and they could pass for error corrections. Fortunately, it is not very likely that individual style changes will turn out to be as frequent as to alter the top overall results.

5. Integrating the error corpus with other corpora

How to make an error corpus compatible with the rest of Polish corpora? Using the same (or compatible) POS tag set seems to be the basic level of interoperability. In other words, the corpus should be tagged with the same or similar tagger. In the tests, an open source tagger compatible with the existing IPI PAN Polish corpus (see Przepiórkowski 2004) was used.
Another level of compatibility is using the same query language to process the corpus. The existing IPI PAN Polish Corpus is using Poliqarp, an efficient corpus query engine (Janus and Przepiórkowski 2007). The projected Polish National Corpus will probably use Poliqarp, at least as one of the options. Poliqarp query language, especially with projected statistical extensions, seems very well suited for the task of processing the error corpus.
Nevertheless, there are serious limitations in Poliqarp: it cannot use any XML files compatible with XCES (Ide, Bonhomme, Romary 2000) and is limited to a specific IPI PAN DTD. Using standard ways of encoding edits in the corpus (like TEI tags *corr* and *gap*) was therefore impossible. That is why for compatibility with current IPI PAN corpus, a pseudo-syntactic tags “ins” and “del” were introduced in the experimental version of Poliqarp-encoded error corpus portion (only a part of the 3GB corpus was processed as a proof of concept). This way, it is possible to use standard Poliqarp statistic query such as *[pos=”del”], sort by freq* to get the sorted frequency list of deleted words from the corpus.
In principle, one could also try to encode corrections and deletions in the metadata part for a chunk stored in the corpus. This does not seem practical as it makes queries harder. Another possibility, which is using a parallel corpus to store revisions, is not only currently not supported by Poliqarp but also superfluous given the fact that standard TEI specification already contains standard markup for deleted and inserted tokens.
Although the experimental Poliqarp version of the corpus uses simply additional tags, it seems to play its role sufficiently. It could be used for storing an error corpus; additionally, it should be able to color the text accordingly (or use underline, overstrike attributes etc.)
In the future, the Polish error corpus should contain not only Wikipedia material but also a sample of learner's results, a sample of proofreaders corrections, and a sample of some other material. This way the error corpus could be used for discovering common implicit knowledge about norms

functioning in the language. And this knowledge can be used for justifying the correctness common, though previously criticized, forms, and – at the same time – for developing grammar checkers dealing with unacceptable forms.

REFERENCES

Adams, R. and Wu, M. (2002). *PISA 2000: Technical report*. Paris: OECD.

Ide, N., Bonhomme, P., & Romary, L. (2000). *XCES: An XML-based Encoding Standard for Linguistic Corpora*. In *Proceedings of the Second International Language Resources and Evaluation Conference*. Paris: European Language Resources Association.

Janus, D. & Przepiórkowski, A. (2007). *POLIQARP 1.0: Some technical aspects of a linguistic search engine for large corpora*, in: Waliński, J., Kredens, K., Góźdź-Roszkowski, S. (ed.), Corpora and ICT in Language Studies. PALC 2005, Frankfurt am Main: Peter Lang.

Klemensiewicz, Z. (1966). *Poprawność i pedagogika językowa*, in: Urbańczyk, S. (ed.), *Polszczyzna piękna i poprawna*, Wrocław, Warszawa, Kraków, Gdańsk: Zakład Narodowy im. Ossolińskich.

Pisarek, W. (1978). *Słownik języka niby-polskiego, czyli błędy językowe w prasie*. Wrocław, Warszawa, Kraków, Gdańsk: Zakład Narodowy im. Ossolińskich.

Przepiórkowski, A. (2004). *Korpus IPI PAN. Wersja wstępna / The IPI PAN Corpus: Preliminary version*. Warszawa: IPI PAN.

Sågvall Hein, A. (1998). *A Chart-Based Framework for Grammar Checking Initial Studies*. In *NODALIDA '98 Proceedings Vol. 11*. Denmark: Center for Sprogteknologi.