Scratching the Surface of Possible Translations

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{bojar,machacek,tamchyna,zeman}@ufal.mff.cuni.cz

Abstract. One of the key obstacles in automatic evaluation of machine translation systems is the reliance on a few (typically just one) human-made reference translations to which the system output is compared. We propose a method of capturing millions of possible translations and implement a tool for translators to specify them using a compact representation. We evaluate this new type of reference set by edit distance and correlation to human judgements of translation quality.

Keywords: machine translation, evaluation, reference translations

1 Introduction

The relationship between a sentence in a natural language as written down and its meaning is a very complex phenomenon. Many variations of the sentence preserve the meaning while other superficially very small changes can distort or completely reverse it. In order to process and produce sentences algorithmically, we need to somehow capture the semantic identity and similarity of sentences.

The issue has been extensively studied from a number of directions, starting with thesauri and other lexical databases that capture synonymy of individual words (most notably the WordNet [13], [14]), automatic paraphrasing of longer phrases or even sentences (e.g. [2], [11], [9]) or textual entailment [1]. We are still far away from a satisfactory solution.

The field of machine translation (MT) makes the issue tangible in a couple of ways, most importantly within the task of automatic MT evaluation. Current automatic MT evaluation techniques rely on the availability of some reference translation, i.e. the translation as produced by a human translator. Obtaining such reference translations is relatively costly, so most datasets provide only one reference translation, see e.g. [8].

Figure 1 illustrates the situation: while there are many possible translations of a given input sentence, only a handful of them are available as reference translations. The sets of hypotheses considered or finally selected by the MT system can be completely disjoint from the set of reference translations. Indeed, only about 5–10% of reference translations were *reachable* for Czech-to-English translation [4], and about a third of words in a system output are not confirmed by the reference despite not containing any errors based on manual evaluation [5]. Relying mostly on unreachable reference translations is detrimental for MT system development. Specifically, automatic MT evaluation methods perform worse and consequently automatic model optimization suffers.

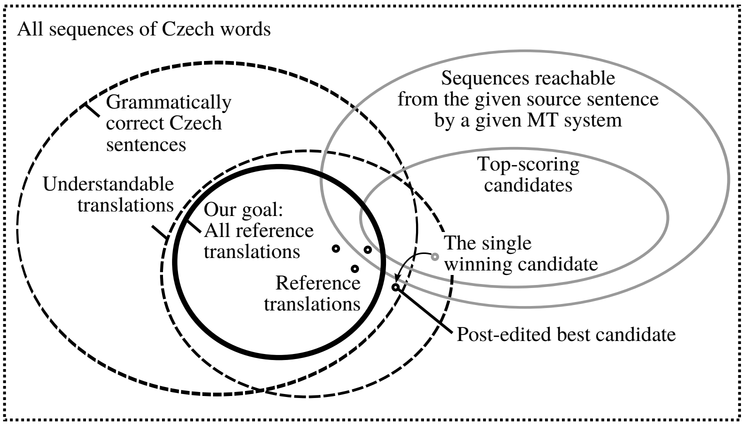
All sequences of Czech words
Grammatically correct Czech sentences
Understandable translations
Our goal: All reference translations
Reference translations
Sequences reachable from the given source sentence by a given MT system
Top-scoring candidates
The single winning candidate
Post-edited best candidate

Fig. 1. The space of all considerable translations of a given source sentence. Human-produced sets are denoted using black lines, machine output is in grey.

We would like to bring the sets of acceptable and reachable translations closer to each other, providing more space for optimal hypothesis selection. This paper presents one possible step in that direction, namely significantly enlarging the set of reference translations. As outlined above, the dataset we created could serve well in research well beyond the MT field, e.g. in an analysis of *sentence-level* paraphrases.

In Section 2, we describe our annotation tool for producing large numbers of correct translations and relate it to a similar tool developed for English [10]. Section 3 provides basic statistics about the number of references we collected and Section 4 carefully analyzes and discusses their utility in MT evaluation.

2 Annotation Tools for Producing Many References

The most inspiring work for our experiment was that of Dreyer and Marcu [10]. Their annotators produce "translation networks", a compact representation of many references, to be used in their HyTER evaluation metric.

We experimented with their annotation interface developed primarily for English but found it rather cumbersome for Czech and other languages with richer morphology and a higher degree of word order freedom.

2.1 Recursive Transition Networks

The approach of [10] is based on recursive transition networks (RTN, [16]), a formalism with the power of context-free grammars. The main building block of the annotation tool in [10] is called a "card" and it covers multiple translations of a short phrase. By combining cards, a large network for the whole sentence can be built. Every path through the network represents a new sentence and the annotators construct the networks so that all such sentences are synonymous. See Figure 2 for an example.

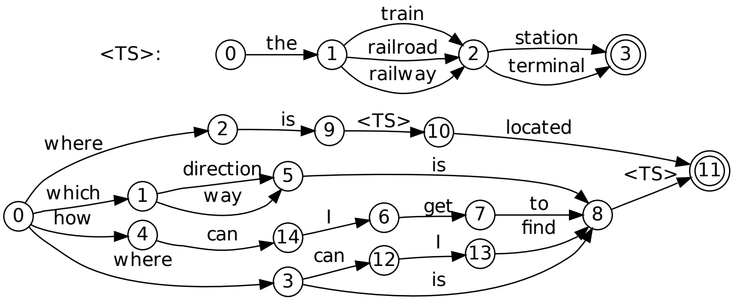

Fig. 2. An example of English recursive transition network from [10].

Reordering of phrases (not discussed in [10]) is possible within the RTN framework by changing the order of cards. The need of such a mechanism in English may seem negligible, yet there are situations such as direct speech where mutual positions of large blocks of text are perfectly interchangeable; similar patterns occur in Czech as well:

- He wanted to step down, he said, “so I could work with more freedom.”
- He said: “I want to step down so I could work with more freedom.”
- “I want to step down so I could work with more freedom,” he said.

More importantly, it is difficult to specify *conditions* under which particular cards can combine in RTN. In morphologically rich languages, we often have to translate a phrase differently based on morpho-syntactic rules. For instance, functions of English verb arguments are determined using word order and prepositions. In Czech, they are determined using prepositions and morphological cases. Verbs subcategorize for noun phrases in particular cases. Consequently, if a verb is replaced by a synonym (or if the verb phrase is passivized or nominalized), the required case for the arguments of the verb may change. The case of the noun must then be reflected by its adjectival modifiers.

Let us illustrate this by an example. Consider the sentence *“The city council approved a new regulation.”* The cards that would model pieces of this sentence in English could be combined more or less freely (within the fixed word order):[1]

the (city council / local government) (approved / gave blessing to / agreed with) a new (regulation / directive / decree)

If we ignored morphology, we could get a very similar picture with the Czech equivalents of the phrases:

(městská rada / zastupitelstvo města) (schválila / požehnala / souhlasila s) nový (předpis / směrnici / nařízení)

So far the alternatives within each part of the sentence differ *lexically*. However, the lexical selections have morphological implications and thus we also have to define *morphological* alternatives:

[1] We are aware that certain domains are much more sensitive to meaning distortions (directive vs. regulation) or inappropriate register (approve vs. give blessing). Our definition of meaning equivalence is not strict. We permit slight divergence if it can be reasonably expected that a human translator will pick either of the alternatives.

- One subject is feminine *(městská rada)*, the other is neuter *(zastupitelstvo města)*. Their gender must be reflected by the verbs *(schválila / schválilo // požehnala / požehnalo // souhlasila s / souhlasilo s)*.
- On a similar note, the three synonyms for *regulation* differ in gender, which dictates different suffixes for the adjective *new: nový předpis / novou směrnici / nové nařízení*.
- Each of the three verbs subcategorizes for a different case: *schválila* requires object in accusative, *požehnala* in dative and the preposition in *souhlasila s* requires instrumental. Thus we have *nový předpis / novému předpisu / novým předpisem // novou směrnici / nové směrnici / novou směrnicí // nové nařízení / novému nařízení / novým nařízením*.

2.2 Unification-Based Annotation

The RTN framework gives us a powerful tool to *combine* “cards”. We would ideally want a tool that also lets us specify the *constraints* that must be fulfilled if two cards are to be combined.

We thus created our own compact representation for languages similar to Czech. Our main building block called *bubble*, comparable to the cards of [10], is defined by:

- the set (possibly discontinuous) of source language tokens it covers;
- the set of conditions it meets;
- the set of translation alternatives in the target language. Every alternative in the set covers the same set of source tokens and meets the same conditions.

A translation alternative is composed of *atoms* (tokens of the target language) and/or *slots* (positions in the translation alternative, specifying properties of other bubbles that are permitted to fill the slot). Where an RTN would refer to a smaller transition network (card) by its name, we refer to a smaller bubble by enumerating the constraints it must meet. For instance, we ask that the bubble covers the source word *regulation* (it may cover more words but this one must be among them) and that its form is in the accusative. Obviously we could achieve the same result in RTN by using more explicative card names, e.g. *regulation-acc*. Our approach is equivalently expressive but it increases annotators’ comfort as well as maintainability of the whole system. While RTNs could be rewritten as a context-free grammar, our approach can be thought of as a unification grammar.

Typical creation of a translation network is analogous to traversing the dependency tree structure that models the syntax of the sentence. One starts at the verb, defines its translations and creates slots for its arguments (and adjuncts). Each argument typically receives its own bubble. The bubble can define alternations for a whole noun phrase, or it can again use slots to separate description of a modifier that could be reused elsewhere. Occasionally a bubble represents a subordinated clause and the process is applied recursively.

Unlike in common unification grammars, we are not forced to annotate a full syntactic tree. It is possible e.g. to create one flat bubble for the whole sentence. The only guiding principle in creating nested bubbles is economy: a set of alternations useful at two or more places is a candidate for a new bubble.

Along the same lines, the decomposition of the sentence into bubbles does not need to reflect linguistic constituents. Sometimes it is practical to take punctuation as the root, rather than the verb; high-level word order decisions drive the distribution of commas and quotation marks around clauses; co-ordination could be preferred over dependency etc. The set of possible constraints is not restricted in any way (e.g. to morphological categories and their values). The annotator is free to introduce arbitrary constraints, e.g. for ensuring good co-reference patterns, auxiliaries in coordination, rhematizers and negation interplay or even style and register features.

We developed two annotation environments in which translators create the compact representations. The Prolog programming language appears to be ideally suited for evaluation of constraints and expansion of bubbles. Several translators encode their annotations directly in Prolog. For those less technically capable we also designed a web-based graphical interface.

2.3 Prolog Interface

Roughly 300 lines of pre-programmed Prolog code provide the necessary set of predicates that check constraints (bubble-slot compatibility) and make sure that all tokens of the source sentence are covered. The translator essentially creates a set of clauses for the predicate `option()`, each of those encodes a bubble. Every `option` lists the source words covered, the conditions met and the target sequence consisting of atoms and slots. A slot refers at least to one source word that must be covered by the bubble in the slot; optionally, it also specifies additional conditions that must be met. Example:[2]

```
% option(+SrcWordsCovered, +ConditionsMet, +OutputAtomsAndSlots).
option([the, city, council], [], [městská, rada, [approved, fem]]).
  % "The city council" can be translated as "městská rada" and then
  % it requires the translation of "approved" in feminine gender.
option([approved], [fem], [schválila, [regulation, acc]]).
option([approved], [fem], [souhlasila, s, [regulation, ins]]).
  % Different translations of "approved" require "regulation" in
  % either "acc"usative or "ins"trumental cases.
option([a, new, regulation], [acc], [nový, předpis]).
option([a, new, regulation], [ins], [novým, předpisem]).
option([a, new, regulation], [acc], [novou, směrnici]).
option([a, new, regulation], [ins], [novou, směrnicí]).
```

A few additional constructs such as the logical `or()` and the possibility to simply drop a token further expand the tools the translators have at their disposal; note however that the syntax that the annotators have to grasp is extremely simple and virtually no knowledge of programming in general or Prolog in particular is required.

2.4 Web Interface

Some translators will be scared by any programming language, regardless how easy it may be. Others may face technical issues regarding installation and running a Prolog

[2] A pre- and post-processor enables using uppercase letters and non-English letters freely in the `option()` predicates.

interpreter on their laptops. In order to accommodate all translators, we also developed a web-based graphical interface. It works as a wrapper for the Prolog engine: bubbles defined in the browser are sent to the server, converted to Prolog clauses and evaluated. The server then sends back either the full list of translations (which is only practical for shorter sentences) or the list of differences against the previous state.

The main problem of the web interface was that we did not anticipate that many bubbles and translations generated. The tool implemented is thus too heavy in terms of processing time, network load and even the required screen space.

3 Collected Data

We selected 50 sentences from the WMT11 test set [7] for our annotation. This particular test set was used in various experiments before and we can thus use:

- manual system rankings of the official WMT11 manual evaluation (see also [3] for a discussion of the rankings)
- the single official reference translation of WMT11 (denoted “O” in the following)
- three more reference translations that come from the German version of the test set [6] (denoted “G” in the following)
- two manual post-edits of a phrase-based MT system similar to those participating in the WMT11 competition (denoted “P”; this set contains only 1997 sentences, each post-edited by two independent annotators.)

Six translators were involved in the task, producing 77 sets of references altogether. Of the 50 sentences, 24 were translated by one annotator, 25 by two and one sentence has three annotations. Each annotator was instructed to spend at most 2 hours translating one sentence. More than 1 hour was needed for a typical sentence.

Utilizing the existing versions, one of the translators used German as the input language; the others used English. All of the annotators had access to one pre-existing human translation in English, German, Spanish and French, and up to four Czech translations, which they could use for inspiration.[3]

We combine networks produced by different translators using simple union of the sets of target sentences (*finite-state union* of [10]). This set is denoted “D” in the following.

Table 1 shows basic statistics of the annotations. In all cases, annotators who used the Prolog interface were more productive, creating over 255 thousand reference translations per sentence on average, compared to roughly 49 thousand references produced by the web users.

For the 25 sentences annotated by two translators, we measured the overlap between the sets of references. We created the union and intersection of the two sets (after tokenizing the sentences) and simply measured the ratio. The results provide further evidence of the richness of possible translations – on average, the overlap was only 4.4% of the produced references. With the exception of two very short sentences, the overlap was always below 10%. See Figure 3 for some examples.

[3] The texts are news articles of mixed origin. One of the pre-existing “translations” was actually the original text.

Table 1. Basic statistics of our annotations.

Annotator	Interface	# of Sents.	Avg. Sent. Length	Avg. Number of Refs.
A	Prolog	3	14.0	483k
B		5	22.5	246k
C		20	24.1	223k
D	Web	25	25.1	54k
E		19	23.0	26k
F		5	15.7	111k

Great when a film has several target groups , but a shame if they are mutually exclusive .

Je výborné , když má film větší počet cílových skupin , jen je smutné , že se navzájem vylučují .
Není k zahození , pokud má film vícero cílových skupin , jen je smůla , když se navzájem vylučují .
Skvělé , že má film víc cílových skupin , je ale politováníhodné , pokud se navzájem neslučují .
Prima , pokud má film více cílových skupin , jen je smutné , pokud se vzájemně neslučují .

Je výhodné , když snímek má několik cílových skupin , smutné ale je , pokud jsou navzájem neslučitelné .
Skvělé je , když snímek má několik cílových skupin , škoda ale je , pokud se tyto skupiny vzájemně vylučují .
Dobré je , když má film více cílových skupin , je jen smolné , pokud jsou tyto navzájem neslučitelné .
Výhodou je , pokud se snímek zaměří na několik cílových skupin , je ale smolné , pakliže jsou vzájemně neslučitelné .

Fig. 3. Four random samples from 196k (Web) and 943k (Prolog) variations of one sentence.

4 Analysis

4.1 String Similarity

Our metric of similarity is based on the well-known Levenshtein distance [12] and returns a value between 0 (completely different) and 1 (identical strings):

$$\mathrm{similarity}(x, y) = 1 - \frac{\mathrm{levenshtein}(x, y)}{\max(\mathrm{length}(x), \mathrm{length}(y))}$$

In order to quantify the diversity of the produced manifold translations, we sampled pairs of translations of each sentence, looking for the pair with the smallest similarity. For four sentences, translations with similarity below 0.1 were found in the samples. The minimum similarity averaged over sentences in our dataset was 0.24 ± 0.13 (standard deviation). This result indicates how much the surface realizations differed while preserving identical meaning.

We also measured the string similarity between system outputs and various reference translations. Table 2 summarizes the results. Unsurprisingly, each of the two manually post-edited translations (P_1 and P_2) are much closer to the original system outputs than the official reference translation (O). However, our annotators managed to produce references (D) which are almost exactly as close as the post-edited translations, even though they did not have access to them or the system outputs (the table shows the similarity with the closest reference translation found).

Table 2. String similarity between system outputs and reference translations.

System	String Similarity to the Given Reference			
	D (closest)	O	P_1	P_2
online-B	0.65	0.55	0.66	0.65
cu-tamchyna	0.65	0.52	0.68	0.69
cu-bojar-contrastive	0.65	0.52	0.64	0.66
cu-bojar	0.65	0.51	0.65	0.68
uedin-contrastive	0.64	0.54	0.64	0.65
uedin	0.64	0.54	0.64	0.65
cu-tamchyna-contrastive	0.64	0.50	0.66	0.66
cu-marecek	0.64	0.52	0.67	0.68
jhu-contrastive	0.62	0.52	0.61	0.61
jhu	0.62	0.51	0.59	0.61
cu-zeman	0.61	0.52	0.61	0.62
cu-popel	0.60	0.51	0.59	0.61
commercial1	0.57	0.48	0.57	0.57
commercial2	0.56	0.46	0.57	0.57

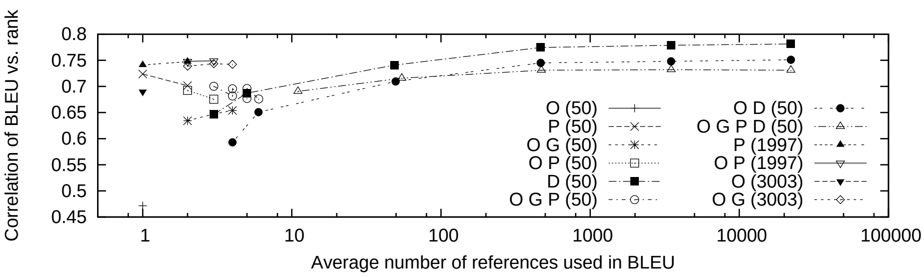

Fig. 4. Correlation of BLEU and manual system rankings with varying sets of references.

4.2 Correlation with Manual Ranking

We evaluate the utility of the manifold reference translations by measuring the Pearson correlation between manual MT system evaluation and the common automatic MT evaluation method BLEU [15]. BLEU was originally tested with 4 reference translations and the number of reference translations is known to strongly influence its performance. In spite of that, BLEU is very often used with just a single reference translation, hoping that a larger test set (more sentences) will compensate the deficiency.

Figure 4 plots Pearson correlation of the official WMT11 system rankings and BLEU when varying the size of the test set (50, 1997 or 3003 sentences as noted in the legend) and the average number of references per sentence. The sources O, P, and G provide us with 1, 2 or 3 references respectively. Our new dataset (denoted “D”) consists of all the references generated from the web or Prolog annotation interface by any of the annotators. The number of references for each sentence differs. We shuffle them and take up to 5, 50, 500, . . . , 50k items from the beginning.

The very baseline correlation is 0.47 (“O 50” in the chart), obtained with only the single official reference translation on the test set reduced to the 50 sentences where we have our extended references. Using the full test set (“O 3003”), the correlation jumps to 0.69. The 1, 2 or 3 additional references coming from German (“O G 3003”) indicate how well the “standard” BLEU should fare: around 0.74. These four references (one official and three coming from German) on the small set of 50 sentences lead to quite a low result: 0.65.

A notable result is 0.72 (“P 50”) obtained on the 50-sentences test set when we use the post-edited translations instead of the official translations. The official translations are obviously more distant from what the systems are capable of producing given the source. With distant references, large portions of output are not scored and systems may differ greatly in the translation quality of those unscored parts. It thus seems sensible to manually post-edit just 50 sentences coming from a baseline version of an MT system and evaluate modifications of the system on this small but tailor-made test set rather than on a larger less-matching set, perhaps even if it had 4 reference translations. The post-edits may however still suffer some problems: in our case, using not just one of the post-edit versions but both of them, the correlation drops to 0.70.

Lines in the chart extending beyond 10 references include our large reference sets. The limit on the number of references has to be taken with caution: subsampling 50 references from a 100k set constructed in 2 hours of annotation is bound to give better results than stopping the construction as soon as it generates 50 options. We nevertheless see that around 5k references, the correlation curves flatten. This could be attributed both to the inherent limitations of BLEU (evaluating the precision of up to 4-grams of words) as well as still the low coverage of the test set. Dreyer and Marcu mention that even billions of references obtained from two annotators still did not include many of the translations suggested by a third annotator.

The solid baseline 0.69 (“O 3003”) is reached shortly after 5 random items from our annotation only (the second point on the line “D 50”), a much smaller set of sentences with many possible translations. Using up to 50k of the possible translations leads to the correlation of 0.78. However, we cannot claim that spending 2 hours × 2 annotators × 50 sentences, i.e. about 200 hours of work, is better than translating 3003 sentences (also about 200 hours of work), because our annotators *did* have access to several versions of each of the sentences.

5 Conclusion

We developed a method and two annotation environments for producing many (possibly all) acceptable translations of a sentence. We performed a small scale experiment in which human translators processed 50 sentences. Obviously this annotation process is very costly (1 to 2 hours per sentence) and we cannot expect anyone to annotate large datasets this way. However, even the small sample provides a completely new point of view for the evaluation of machine translation output (tens or hundreds of thousands of references per average sentence). As expected, automatic evaluation against larger sets of references shows higher correlation with human judgment of translation quality.

A surprising observation is that just 50 post-edited translations serve as an equal or better reference than 3003 independent translations (correlation 0.70–0.72 vs. 0.69).

The annotated data we created is available to the research community. Besides machine translation, it can be also used to evaluate other NLP tasks, ranging from paraphrasing to grammar development or parsing.

6 Acknowledgements

This work was partially supported by the grants P406/11/1499 of the Grant Agency of the Czech Republic, FP7-ICT-2011-7-288487 (MosesCore) of the European Union and 1356213 of the Grant Agency of the Charles University. We are grateful to Markus Dreyer for a preview of their annotation interface.

References

1. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research 38, 135–187 (2010)
2. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proc. of ACL. pp. 597–604. Ann Arbor, Michigan, USA (2005)
3. Bojar, O., Ercegovčević, M., Popel, M., Zaidan, O.: A Grain of Salt for the WMT Manual Evaluation. In: Proc. of WMT. pp. 1–11. ACL, Edinburgh, Scotland (2011)
4. Bojar, O., Kos, K.: 2010 Failures in English-Czech Phrase-Based MT. In: Proc. WMT and MetricsMATR. pp. 60–66. ACL, Uppsala, Sweden (2010)
5. Bojar, O., Kos, K., Mareček, D.: Tackling Sparse Data Issue in Machine Translation Evaluation. In: Proc. of ACL Short Papers. pp. 86–91. ACL, Uppsala, Sweden (2010)
6. Bojar, O., Zeman, D., Dušek, O., et al.: Additional German-Czech reference translations of the WMT'11 test set, `http://hdl.handle.net/11858/00-097C-0000-0008-D259-7`
7. Callison-Burch, C., Koehn, P., Monz, C., Zaidan, O.: Findings of the 2011 Workshop on Statistical Machine Translation. In: Proc. of WMT. pp. 22–64. ACL (2011)
8. Callison-Burch, C., Koehn, P., Monz, C., et al.: Findings of the 2012 Workshop on Statistical Machine Translation. In: Proc. of WMT. pp. 22–64. ACL, Montréal, Canada (2012)
9. Denkowski, M., Lavie, A.: Meteor-next and the meteor paraphrase tables: Improved evaluation support for five target languages. In: Proc. of WMT and MetricsMATR. pp. 339–342. ACL, Uppsala, Sweden (2010)
10. Dreyer, M., Marcu, D.: HyTER: Meaning-Equivalent Semantics for Translation Evaluation. In: Proc. of NAACL/HLT. pp. 162–171. Montréal, Canada (2012)
11. Kauchak, D., Barzilay, R.: Paraphrasing for Automatic Evaluation. In: Proc. of NAACL/HLT. pp. 455–462. New York City, USA (2006)
12. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet Physics-Doklandy. vol. 10 (1966)
13. Miller, G.A.: WordNet: A lexical database for English. Commun. ACM 38(11), 39–41 (1995)
14. Pala, K., Čapek, T., Zajíčková, B., et al.: Český WordNet 1.9 PDT (2010), `http://hdl.handle.net/11858/00-097C-0000-0001-4880-3`
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proc. of ACL. pp. 311–318. Philadelphia, Pennsylvania (2002)
16. Woods, W.A.: Transition network grammars for natural language analysis. Commun. ACM 13(10), 591–606 (1970)