The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011

EDITORIAL BOARD

Published twice a year by Charles University in Prague

Editorial office and subscription inquiries:
ÚFAL MFF UK, Malostranské náměstí 25, 118 00, Prague 1, Czech Republic
E-mail: pbml@ufal.mff.cuni.cz

ISSN 0032-6585

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011

CONTENTS

Articles

Notes

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 5–18

A Guide to Jane, an Open Source Hierarchical Translation Toolkit

Daniel Stein, David Vilar, Stephan Peitz, Markus Freitag, Matthias Huck, Hermann Ney

Chair for Computer Science 6, RWTH Aachen University

Abstract

Jane is RWTH's hierarchical phrase-based translation toolkit. It includes tools for phrase extraction, translation and scaling factor optimization, with efficient and documented programs of which large parts can be parallelized. The decoder features syntactic enhancements, reorderings, triplet models, discriminative word lexica, and support for a variety of language model formats. In this article, we will review the main features of Jane and explain the overall architecture. We will also indicate where and how new models can be included.

1. Introduction

This article describes the open source hierarchical phrase-based decoder Jane and its associated toolkit, which was released for non-commerical use in Vilar et al. (2010). Jane follows the hierarchical phrase model as described in Chiang (2007), which can be seen as an extension of the standard phrase model, where the phrases are allowed to have "gaps". In this way, long-distance dependencies and reorderings can be modelled in a consistent way. As in nearly all current statistical approaches to machine translation, this model is embedded in a log-linear model combination.

Jane features syntactic enhancements, additional reorderings, triplet models, discriminative word lexica, and support for a variety of language model formats. The toolkit also implements algorithms for phrase table extraction, translation and scaling factor optimization. Most processes can be parallelized if the Sun Grid Engine is installed. RWTH has been developing this toolkit during the last two years and it was used successfully in numerous machine translation evaluations. It is written in

 Corresponding author: stein@cs.rwth-aachen.de
Cite as: Daniel Stein, David Vilar, Stephan Peitz, Markus Freitag, Matthias Huck, Hermann Ney. A Guide to Jane, an Open Source Hierarchical Translation Toolkit. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 5–18. doi: 10.2478/v10108-011-0001-6.

C++ with special attention to clean code, extensibility and efficiency, and is available under an open-source non-commercial license.

This article is mainly directed at developers looking for a short overview of the toolkit's architecture. We will briefly review the main features of Jane, with a strong focus on implementation decisions, and we will also describe how and where new extraction and translation models should be implemented by taking advantage of existing classes. For a more general description, we refer to Vilar et al. (2010), and for performance results we refer to system descriptions of international evaluations, e.g. Heger et al. (2010). Note that an in-depth manual is included in the Jane package as well.

The article is structured as follows: we review the tool invocation in Section 2. Then, we review the main features that are implemented and how they can be extended, first for the extraction (Section 3), then for the decoding (Section 4). We proceed to mention some other included tools in Section 5. After a short comparison with Joshua in Section 6, we give a short conclusion in Section 7.

1.1. Related Work

Jane implements features presented in previous work, developed both at RWTH and other groups. As we go over the features of the system we will provide the corresponding references. It is not the first system of its kind, although it provides some unique features. There are other open source hierarchical decoders available, one of them being SAMT (Zollmann and Venugopal, 2006). The current version is oriented towards Hadoop usage, the documentation is however still missing. Joshua (Li et al., 2009, 2010) is a decoder written in Java by the John Hopkins University. This project is the most similar to our own, however both were developed independently and each one has some unique features. Moses (Koehn et al., 2007) is the de-facto standard phrase-based translation decoder and has now been extended to support hierarchical translation.

2. Invocation

The main core of Jane has been implemented in C++. It is mainly directed at linux systems, and uses SCons (`http://www.scons.org`) as its build system. Prerequisites for some tools are the SRI language model (Stolcke, 2002), cppunit and the Numerical Recipes (Press et al., 2002). Upon building, a variety of programs and scripts are created to offer a flexible extraction, translation and optimization framework. Alignment training is not included, since well established tools for this purpose exist. Jane accepts most common alignment formats.

In general, all tools support the option `--help` which outputs a compact description of the available command line options. Some programs also support `--man` for a more verbose description in the form of a Unix man page. These manual pages are

generated automatically and thus are always up-to-date. Nearly all components accept a `--verbosity` parameter for controlling the amount of information they report. The parameter can have 6 possible values, ranging from `noLog` to `insaneDebug`.

The options have a hierarchical structure, and the more complex modules within a larger tool typically have their own naming space. For example, the size of the internal n-best list in the cube prune algorithm is set with `--CubePrune.generationNbest`, and the file for the language model can be set with `--CubePrune.LM.file`. We refer to each of these sections as *components*. There are components for the search algorithms, for the language models, for the phrase table, et cetera. The name can also be replaced by a wildcard (*) if all components are to be addressed.

Although all of the options can be specified in the command line, a config file can be used in order to avoid repetitive typing, by invoking the program with `--config <config-file>`. Options specified in the command line have precedence over the config file.

If the Sun Grid Engine (`http://www.sun.com/software/sge/`) is available, nearly all operations of Jane can be parallelized. For the extraction process, the corpus is split into chunks (the granularity being user-controlled) which are distributed in the computer cluster. Count collection, marginal computation and count normalization all happen in an automatic and parallel manner. For the translation process, a batch job is started on a number of computers. A server distributes the sentences to translate to the computers that have been made available to the translation job. Additionally, for the minimum error rate training methods, random restarts may be performed on different computers in a parallel fashion.

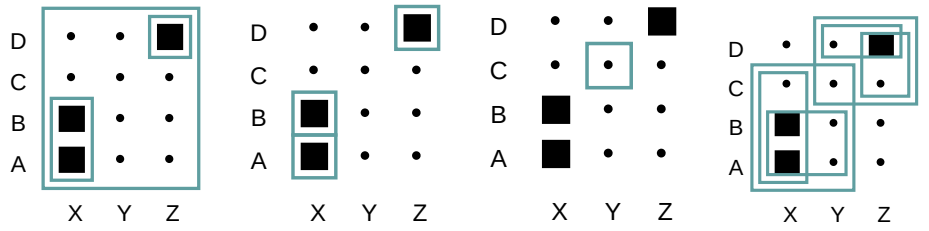
The same client-server infrastructure used for parallel translation may also be reused for interactive systems. Although no code in this direction is provided, one would only need to implement a corresponding frontend which communicates with the translation server (which may be located on another machine).

3. Extraction

In the extraction process, for every training source sentence f_1^J, target sentence e_1^I and alignment $\mathcal{A}$ we generate a set of phrases. First, we extract the set of initial phrases, as defined for the standard phrase-based approach:

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, \mathcal{A}) := \{ \langle f_{j_1}^{j_2}, e_{i_1}^{i_2} \rangle \mid j_1, j_2, i_1, i_2 \qquad & \text{so that} \\ \forall (j,i) \in \mathcal{A} : (j_1 \le j \le j_2 \Leftrightarrow i_1 \le i \le i_2) \qquad & (1) \\ \wedge \exists (j,i) \in \mathcal{A} : (j_1 \le j \le j_2 \wedge i_1 \le i \le i_2) \} . & \end{aligned}$$

See Figure 1(a) for an example of a valid lexical phrase. Words that remain unaligned in the corpus might be problematic for the translation if at translation time they appear in different contexts as in the training corpus. They are not treated as



(a) Valid lexical blocks (b) Single Word Heuristic (c) Forced Single Word Heuristic (d) Extraction of Non-Aligned Word Pairs

Figure 1. Extraction heuristics applied for initial phrases

unknown words, as Jane has them in its internal vocabulary, but is unable to produce a valid parse in the decoding step, which is why we allow for three heuristics in the extraction procedure. In the *single word* heuristic, all single-word pairs derived from each alignment are extracted, even if they would not consist of a valid phrase based on Equation 1 (Figure 1(b)). In the *forced single word* heuristic (Figure 1(c)), all word pairs that are neither covered by a source nor a target alignment are extracted as additional word pairs . Finally, in the *non-aligned heuristic*, all initial phrases are also extended whenever there are non-aligned words on the phrase border (Figure 1(d)).

After the extraction of the lexical phrases, we look for those phrases that contain smaller sub-phrases to extract hierarchical phases. The smaller phrases are suppressed and gaps are produced on the larger ones. For computing probabilities, we compute the counts of each phrase and normalize them, i.e. compute their relative frequencies. Note that the heuristics mentioned above are typically restricted from forming hierarchical phrases, since for some corpora we would produce an arbitrary large number of entries, but this behaviour can be controlled through run-time options. Also due to phrase table size considerations, we typically filter the phrases to only those that are needed for the translation of a given corpus.

Figure 2 shows a schematic representation of the extraction process. In a first pass we extract the bilingual phrases that are necessary for translating the given corpus. For normalization, we still need to compute the marginals for the source and target parts of the phrases. For the source marginals, we can already limit the computation at this stage using the source filter text, but for the target marginals, we do not know in advance which ones will be necessary. Therefore we compute them in a second pass, using the target parts of the bilingual phrases as a target filter.

When parallelizing this operation, the corpus is split into chunks which are distributed in the computer cluster. Count collection, marginal computation and count normalization all happen in an automatic manner.

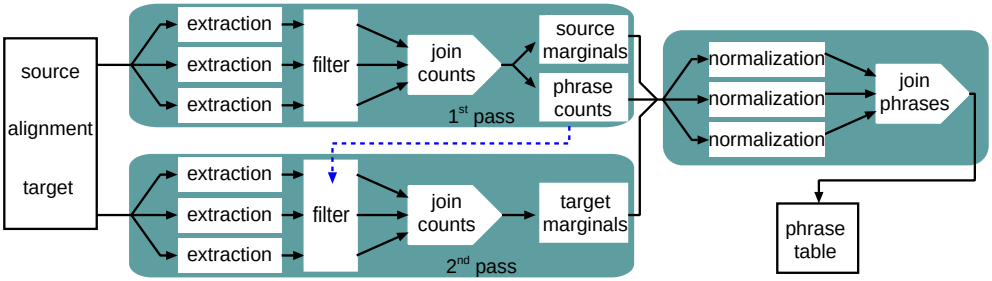


Figure 2. Workflow of the extraction procedure

3.1. Additional Features

Each phrase pair can have an arbitrary number of additional features associated with it. They may hold information about the alignment within the phrase, linguistic information, and more. In order to facilitate the addition of such models, Jane provides two virtual classes, which can be inherited from.

The class `AdditionalExtractionInformation` is a wrapper class for the additional information. The main function that has to be implemented deals with the combination of the feature with another instance of itself. This is necessary whenever the same phrase can be extracted more than once, from different sentence pairs or even in the same sentence pair. Additionally, functions for writing the features into the phrase table have to be provided. In the normalization step, this class is also invoked with the corresponding information about the marginals, in case that it needs this information for producing the final score.

The class `AdditionalExtractionInformationCreator` is responsible for creating instances of AdditionalExtractionInformation. As such, it receives the information available to the extraction process and computes the additional features for the phrase pair. For hierarchical phrases, it may take into account the additional information of the larger phrase and of the phrase that produces the gap in it. In order to assign descriptive labels to the different `AdditionExtractionInformation` classes, new classes must be registered in the corresponding factory functions. The extraction scripts will then automatically perform the necessary steps for generating the additional features, including joining and normalization of the counts.

3.2. Implemented Additional Features

With Jane, it is possible to include numerous additional features in the phrase table. We proceed to review a few of them. The *alignment* information remembers the internal alignment of the extracted phrase. The *dependency* information augments

the phrases with additional dependency-level information in the spirit of Shen et al. (2008). Given a dependency tree at sentence level, we mark phrases that are syntactically valid, and to preserve inner word dependencies. The *heuristic* information marks for each phrase, which of the extraction heuristics as defined in Section 3 have been used to create this particular phrase. The *parse match* information marks whether the phrase matches the yield of a parse tree node, a rather simple approach which has been succesfully applied in Vilar et al. (2008). Finally, the *soft syntactic label* information adds syntactic labels to each phrase and each non-terminal, which are typically derived from a given parse tree. They can be used to compute an additional parse probability based on linguistic experiences, e.g. by emphasizing the need for a verb in the sentence, or by penalizing whenever a noun phrase non-terminal is substituted with a verb phrase, as in Venugopal et al. (2009).

4. Decoding

The search for the best translation proceeds in two steps. First, a monolingual parsing of the input sentence is carried out using the CYK+ algorithm (Chappelier and Rajman, 1998), a generalization of the CYK algorithm which relaxes the requirement for the grammar to be in Chomsky normal form. From the CYK+ chart we extract a hypergraph representing the parsing space.

In a second step the translations are generated from the hypergraph, computing the language model scores in an integrated fashion. Both the cube pruning and cube growing algorithms (Huang and Chiang, 2007) are implemented. For the latter case, the extensions concerning the language model heuristics presented in Vilar and Ney (2009) have also been included.

The majority of the code for both the cube pruning and cube growing algorithms is included in corresponding classes derived from an abstract hypernode class. In this way, the algorithms have access to the hypergraph structure in a natural way. The CYK+ implementation is parametrized in such a way that the derived classes are created as needed. This architecture is highly flexible, and preliminary support for forced alignments in the spirit of Wuebker et al. (2010) is also implemented in this way.

Jane supports four formats for n-gram language models: the ARPA format, the SRI toolkit binary format (Stolcke, 2002), randomized LMs as described in Talbot and Osborne (2007), using the open source implementation made available by the authors of the paper and an in-house, exact representation format. In order to ease the integration of these possibilities, an abstract interface class has been implemented, which provides access to the necessary operations of the language model.

The actual translation procedure is clearly separated from the input/output operations. These are handled in the `RunMode` classes, which are responsible of obtaining the text to translate, calling the translation methods with the appropriate parameters, and writing the result to disk. There are three main RunModes: `SingleBestRunMode` for

single-best operation, `NBestRunMode` for generation of n-best lists and `Optimization-ServerRunMode`. This last one starts a Jane server which offers both single-best and n-best translation functionality, communicating over TCP/IP sockets. In the current implementation this is used for parallel translation and/or optimization in a computer cluster, but it may be easily reused for other applications, like online translation services or interactive machine translation.

For parallelized operation, a series of jobs are started in parallel in the computer cluster. The first one of these jobs is the master and controls the translation process. All Jane processes are started in server mode and wait for translation requests from the master job.

The translation servers are allocated in a dynamic way; if more computers are made available for the translation job, they can be added on the fly to the translation task. The longest sentences are the first ones sent to translate. This simple heuristic performs load balancing, trying to avoid "temporal deadlocks" when the whole array job is just waiting for a computer to finish the translation of a long sentence that happened to be at the end of the corpus. A simple fault-tolerance system is also built-in, which tries to detect if a computer has had problems and resends the associated sentence to another free node. It is however quite basic and although it detects most problems, there are still some cases where non-responding computers may go undetected.

4.1. Additional Models

Jane is designed to be easily extended with new models, added in the log-linear combination. If the new features can be computed at phrase level, the best way is to include them at extraction time, as described in Section 3.1. The decoder can then be instructed to use these additional features at translation time.

For models that cannot computed this way, an abstract `SecondaryModel` class can be derived from. The main function in this class is the `scoreBackpointer` method, which receives a derivation to score, together with a reference to the current hypernode. With this information, the method can obtain all the necessary information for computing the model score. Secondary models implemented this way must be registered in the `SecondaryModelCreator` class. They will be then known to Jane, and facilities for scaling factor allocation, parameter handling and multiple model instantiation will be provided.

There is a limitation to the kind of models that can be implemented this way, namely the models can only influence the search process by generating new scores. No additional information that may change the hypothesis space by e.g. hypothesis recombination is (yet) supported.

4.2. Implemented Additional Models

Like with the additional extraction models, several additional models during the translation are already implemented in Jane using the SecondaryModel infrastructure. In this section we list some of them.

Extended Lexicon Models The Extended Lexicon Models include discriminative lexicon models and triplet models as in Mauser et al. (2009), and are able to take long range dependencies across the whole source sentence into account. The *triplet* model extends the well-known IBM model 1 (Brown et al., 1993), by estimating the probability $p(e|f, f')$ of a target word e based on two source words f, f'. Like IBM 1, the triplet model is trained iteratively using the EM algorithm. During extraction and decoding, f is the source word aligned to the target word e to be translated, while f' ranges over the words in the source sentence. Thus, the second source word f' enables the model to make more informed decisions about translating f into e.

The *discriminative word lexicon* uses the whole source sentence to predict target words, thus taking into account global dependencies. It is modeled as a combination of simple classifiers for each word e from the target vocabulary V_E. Each of these classifiers models whether a certain word e is present in the target sentence ($\delta_e = 1$) or not ($\delta_e = 0$), given the set of source words $\mathbf{f}$. The probability of the target sentence is then modeled as the product of all positive classification probabilities, over all words in the target sentence, times the product of all negative classification probabilities over all words not contained in the target sentence.

Soft Syntactic Labels The Soft Syntactic Labels (cf. Section 3.2) extend the hierarchical model in a similar way as in the work of Venugopal et al. (2009): for every rule in the grammar, we store information about the possible non-terminals that can be substituted in place of the generic non-terminal X, together with a probability for each combination of non-terminal symbols (cf. Figure 3).

During decoding, we compute two additional quantities for each derivation d. The first one is denoted by $p_h(Y|d)$ (h for "head") and reflects the probability that the derivation d under consideration of the additional non-terminal symbols has Y as its starting symbol. This quantity is needed for computing the probability $p_{syn}(d)$ that the derivation conforms with the extended set of non-terminals.

Dependency With the Dependency model, we are able to introduce language models that span over longer distances than shallow n-gram language models. In Figure 4, we can for example evaluate the left-handed dependency of the structure "In", followed by "industry", on the structure "faced". For this, we employ a simple language model trained on dependency structures and compute the probability for the trigram "In industry faced-as-head".

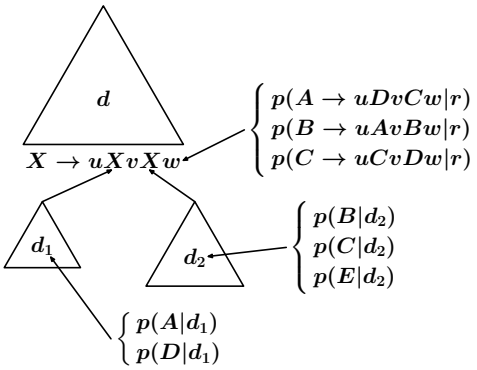


Figure 3. Visualization of the soft syntactic labels approach. For each derivation, the probabilities of non-terminal labels are computed.

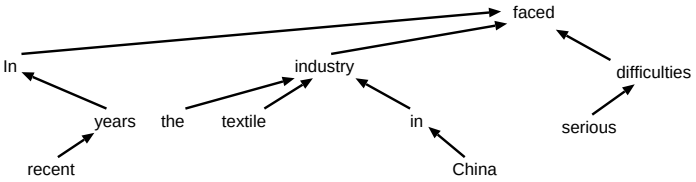


Figure 4. Dependency parsing for the sentence "In recent years, the textile industry in China faced serious difficulties".

The model requires a given dependency tree while extracting the phrase table, and works with this information to derive a tree of the output translation. Note that Shen et al. (2008) only allow for two structures to be extracted: the first possibility is what the authors called a *fixed* dependency structure. With the exception of one word within this phrase, called the *head*, no outside word may have a dependency within this phrase. Also, all inner words may only depend on each other or on the head. For a second structure, called a *floating* dependency structure, the head dependency word may also exist outside the phrase. We do not restrict our algorithms to fixed and floating structures, but rather mark invalid phrases during reconstruction and proceed to reconstruct as much of the dependencies as possible.

Table 1. Speed comparison Jane vs. Joshua.

System	words/sec
Joshua	11.6
Jane cube prune	15.9
Jane cube grow	60.3

5. Other Tools

For optimization of the log-linear scaling factors, we support the minimum error rate training (MERT) described in Och (2003), the MIRA algorithm, applied for machine translation in Chiang et al. (2009), and the downhill simplex method (Nelder and Mead, 1965).

Jane provides a tool to compute the *grow-diag* alignment as presented in Koehn et al. (2003), as well as its alternative as presented in Och and Ney (2003).

6. Comparison with Joshua

As stated in Section 1.1, Joshua (Li et al., 2009) is the most similar decoder to our own, developed in parallel at the Johns Hopkins University.

Jane was started separately and independently. In their basic working mode, both systems implement parsing using a synchronous grammar and include language model information. Each of the projects then progressed independently, and each has unique extension. Efficiency is one of the points where we think Jane outperforms Joshua. One of the reasons can well be the fact that it is written in C++ while Joshua is written in Java. We performed a control experiment on the IWSLT'08 Arabic to English translation taks, using the same settings for both decoders and making sure that the output of both decoders was identical[1]. The speed results can be seen in Table 1. Jane operating with cube prune is nearly 50% faster than Joshua and the speed difference can be improved by using cube growing, although with a slight loss in translation performance. This may be interesting for certain applications like e.g. interactive machine translation or online translation services, where the response time is critical and sometimes even more important than a small (and often hardly noticeable) loss in translation quality.

For comparison of translation results, we refer to the results of the last WMT evaluation shown in Table 2. Johns Hopkins University participated in this evaluation using Joshua, the system was trained by its original authors (Schwartz, 2010) and thus can be considered to be fully optimized. RWTH also participated using Jane among

[1] With some minor exceptions, e.g. unknown words.

Table 2. Results for Jane and Joshua in the WMT 2010 evaluation campaign.

	Jane		Joshua	
	BLEU[%]	TER[%]	BLEU[%]	TER[%]
German-English	21.8	69.5	19.5	66.0
English-German	15.7	74.8	14.6	73.8
French-English	26.6	61.7	26.4	61.4
English-French	25.9	63.2	22.8	68.1

other systems. A detailed description of RWTH's submission can be found in Heger et al. (2010). The scores are computed using the official Euromatrix web interface for machine translation evaluation[2].

As can be seen the performance of Jane and Joshua is similar, but Jane generally achieves better results in BLEU, while Joshua has an advantage in terms of TER. Having different systems is always enriching, and particularly as system combination shows great improvements in translation quality, having several alternative systems can only be considered a positive situation.

7. Conclusion

In this work, we described how and where new models can be integrated into the Jane architecture. We also reviewed the features that are currently implemented. Jane can be downloaded from `http://www.hltpr.rwth-aachen.de/jane`. The toolkit is open-source and free for non-commercial purposes. Other licenses can be negotiated on demand. It is our hope that by adhering to strict code and documentation policies, we enable fellow researchers to adopt and extend the toolkit easily to their needs.

Bibliography

Brown, Peter F., Stephan A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993.

Chappelier, Jean-Cédric and Martin Rajman. A Generalized CYK Algorithm for Parsing Stochastic CFG. In *Proc. of the First Workshop on Tabulation in Parsing and Deduction*, pages 133–137, Apr. 1998.

Chiang, David. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2): 201–228, June 2007.

[2] `http://matrix.statmt.org/`

Chiang, David, Kevin Knight, and Wei Wang. 11,001 new Features for Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 218–226, Boulder, Colorado, June 2009.

Heger, Carmen, Joern Wuebker, Matthias Huck, Gregor Leusch, Saab Mansour, Daniel Stein, and Hermann Ney. The RWTH Aachen Machine Translation System for WMT 2010. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 93–97, Uppsala, Sweden, July 2010.

Huang, Liang and David Chiang. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 144–151, Prague, Czech Republic, June 2007.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology, North American Chapter of the Association for Computational Linguistics*, pages 54–60, Edmonton, Canada, May 2003.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180, Prague, Czech Republic, June 2007.

Li, Zhifei, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. Joshua: An Open Source Toolkit for Parsing-Based Machine Translation. In *Proc. of the Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March 2009. Association for Computational Linguistics.

Li, Zhifei, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Ziyuan Wang, Jonathan Weese, and Omar F. Zaidan. Joshua 2.0: A toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 133–137, Uppsala, Sweden, July 2010.

Mauser, Arne, Saša Hasan, and Hermann Ney. Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 210–218, Singapore, Aug. 2009.

Nelder, John A. and Roger Mead. The Downhill Simplex Method. *Computer Journal*, 7:308, 1965.

Och, Franz Josef. Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July 2003.

Och, Franz Josef and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, Mar. 2003.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK, 2002.

Schwartz, Lane. Reproducible Results in Parsing-Based Machine Translation: The JHU Shared Task Submission. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 177–182, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Shen, Libin, Jinxi Xu, and Ralph Weischedel. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 577–585, Columbus, Ohio, June 2008.

Stolcke, Andreas. SRILM – an Extensible Language Modeling Toolkit. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, volume 3, Denver, Colorado, Sept. 2002.

Talbot, David and Miles Osborne. Smoothed Bloom Filter Language Models: Tera-scale LMs on the Cheap. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 468–476, Prague, Czech Republic, June 2007.

Venugopal, Ashish, Andreas Zollmann, N.A. Smith, and Stephan Vogel. Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 236–244, Boulder, Colorado, June 2009.

Vilar, David and Hermann Ney. On LM Heuristics for the Cube Growing Algorithm. In *Proc. of the Annual Conf. of the European Association for Machine Translation (EAMT)*, pages 242–249, Barcelona, Spain, May 2009.

Vilar, David, Daniel Stein, and Hermann Ney. Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 190–197, Waikiki, Hawaii, Oct. 2008.

Vilar, David, Daniel Stein, Matthias Huck, and Hermann Ney. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proc. of the Workshop on Statistical Machine Translation*, pages 262–270, Uppsala, Sweden, July 2010.

Wuebker, Joern, Arne Mauser, and Hermann Ney. Training phrase translation models with leaving-one-out. In *48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484, Uppsala, Sweden, 2010.

Zollmann, Andreas and Ashish Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, New York, June 2006.

Address for correspondence:
Daniel Stein
stein@cs.rwth-aachen.de
Chair for Computer Science 6
RWTH Aachen University
Ahornstr. 55
52056 Aachen, Germany

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 19–32

Towards a New Approach for Disambiguation in NLP by Multiple Criterian Decision-Aid

Youssef Hoceini[a], Mohamed A. Cheragui[a], Moncef Abbas[b]

[a] High school of Computer Science, Algeria
[b] USTHB University, Algeria

Abstract

The aim of this paper is to present a combination of NLP and Multiple Criteria Decision-Aid (MCDA) in order to reach an effective analysis when dealing with linguistic data from various sources. The coexistence of these two concepts has allowed us, based on a set of actions and criteria, to develop a coherent system that integrates the entire process of textual data analysis (no-voweled Arabic texts) into decision making in case of ambiguity. Our solution is based on decision theory and an MCDA approach with a TOPSIS technique. This method allows the multi-scenario classification of morphosyntactical ambiguity cases in order to come out with the best performance and reduce the number of candidate scenarios.

1. Introduction

In the Arabic language, the duality between the word and vowels[1] implies a large increase in tidal volume of the tongue, knowing that a word can sometimes take more than twenty forms depending on the configuration that accompanies it. In fact, it leads to the most complex problems in understanding humans and machines Hoceini and Abbas (2009a). The phenomenon that arises from this multiplicity is called ambiguity. The determination of a unique morphosyntactic category for each word in the text of a treaty, for instance, is necessary for vowels in the text, and resolves most issues related to automatic processing of Arabic. The specific context of Arabic emphasizes

[1] Consider a set of codes that provide a number of functions have diacritical marks placed above or below the letters appear in some texts as: the Quraan, Hadith, poetry and textbooks in particular.

Corresponding author: y_hoceini@yahoo.fr
Cite as: Youssef Hoceini, Mohamed A. Cheragui, Moncef Abbas. Towards a New Approach for Disambiguation in NLP by Multiple Criterian Decision-Aid. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 19–32. doi: 10.2478/v10108-011-0002-5.

the presence of a multitude of criteria that reflect the function of several constraints (e.g., grammar, semantics, logic and statistics). Therefore, a proper parsing system is required to be robust, fast, and most importantly less ambiguous.

This paper is organized as follows. First, an overall presentation of our morphological analyzer is given with a brief and comprehensive description of the phenomenon of ambiguity. The second part, we deals with the approaches for ambiguity removal or disambiguation. Next, the proposed model is presented along with the aggregation method known as "TOPSIS"[2] and the weighting method called "Entropy". Then, we show the implementation of our model. Finally, we summarize our findings in the conclusion.

Contrary to probabilistic and constraint based rules models, the proposed model of morphosyntactic disambiguation of Arabic implements an original method base on decision theory as an approach to categorize multi scenarios disambiguation in order to bring out the best. This approach has the advantage of reducing dominated scenarios and ranking the rest by different criteria evaluation.

2. Morphological Analysis

The morphological processing of the morpheme is based on two key concepts; The synthesis step that generates words or phrases based on a set of derivation rules, and inflectional adaptations, and the analysis step that associates a word graph to a set of information that describe the morphological and grammatical units of their composition (proclitic, prefix, basic, suffix, enclitic). This information allows the morphological analysis phase to determine the morphological properties of a word, such as: category (or part of speech: verb, noun or article), gender (male or female), number (singular or plural), voice (active or passive), time of action (accomplished or fulfilled), mode of the verb (indicative, subjunctive), and person (first, second or third person).

At this stage, the morphological ambiguity occurs when the analysis assigns a word more than one set of information (or the vice versa), which generates a combinatorial notion. Thus, prior to parsing, we must remove the ambiguity of many morphological labels that are associated to one word.

3. Disambiguation

Disambiguation is a crucial step in the process of morphological analysis. The morphological ambiguity in Arabic is mainly caused by the absence of vowels. According to Debili et al. (2002), 43.03 % of words are ambiguous in the Arabic voweled text. This proportion increases to 72.03 % when the text is not voweled. To sum up, the absence of these signs generates more cases of morphological ambiguity; for instance, the word with no vowels كتب (writing) may have 16 possible vowels, which

[2]TOPSIS: Technique for Order Preference by Similarity to Ideal Solutions

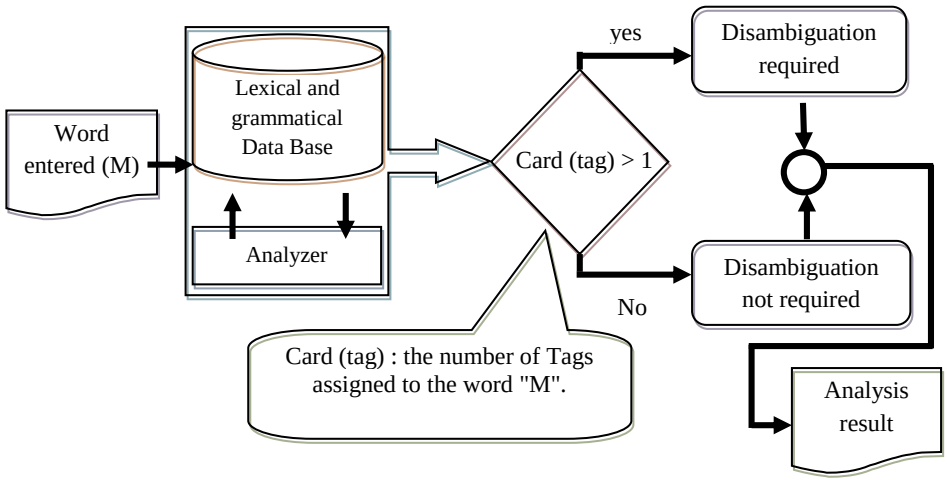


Figure 1. Different disambiguation techniques

leads to 9 different grammatical categories Aloulou et al. (2004). However the phase of disambiguation is not always required in the analysis process. The disambiguation module intervenes if the word receives more than one tag, which generates a situation of confusion or ambiguity (see Figure 1).

3.1. Existing Approaches to Disambiguation

Current analyzers are classified according to their mode of disambiguation. Yet, they all fall into two model classes; the probabilistic models that are meant grammatical labeling, and the constraint models Hoceini and Abbas (2009c). A summary of the different disambiguation techniques is given in Figure 2.

3.1.1. The Constraint Approach

This approach is based on a model that involves a linguist, which will allow the establishment of list of rules per class or category in order to be able to disambiguate. These categories can be: grammatical, structural, semantic, logical, etc… The grammatical constraints are mainly used for removing the ambiguity due to the simultane-

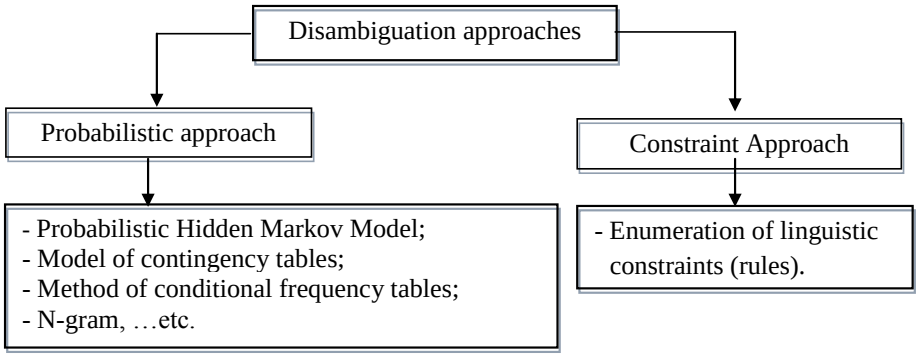


Figure 2. Different disambiguation techniques

ous membership of the semantic unit to more than one grammatical model. The use of grammatical constraints may be sufficient by itself, but sometimes other semantic constraints are imposed.

3.1.2. The Probabilistic Approach

In this approach, the probabilistic and statistical factor classifies constraints based on their redundancy. This is done on the basis of the highest rate of presence of a language constraint which can be lexical, morphological, syntactic, morphosyntactic or semantic. The statistical and probabilistic constraint are determined by searching in the language (corpus) to assess the rate of occurrence of each constraint in relation to other constraints. This rate is estimated using complex arithmetic. The removal of ambiguity is performed using two types of information: the words label and the contextual syntax. Then one proceeds to a combination of both information and learning[3] on their corpus annotated on hand. The Markov technique is a probabilistic model commonly used due to its efficiency Merialdo (1994).

[3]The technique of learning and classification: A set of examples is stored in memory, each set contains a word or its lexical representation, its context (anterior and posterior) and its grammatical category that is related to the context. The analysis is done as follows: for each word in the sentence, the Tager will look for a stored similar example (in memory) and deduce its grammatical category.

3.1.3. Comparison

Many researchers have found that constraint analyzers are faster and easier to implement than the stochastic parsers. In addition, they are more reliable and efficient in terms of analysis. Allotti and Ponsard (2005); Chanod and Tapanainen (1995). A third class of analyzers that combines the two previous approaches is added to increase performance and analysis suitability.

4. Proposed Approach : Multi-criteria Analysis Model

The NLP has frequent decision-making practices that meet a series of choices. Knowing the context of a specific language such as Arabic emphasizes the presence of criteria that reflect the function of several constraints (e.g., grammatical inflectional, structural, semantic, logical and statistics). So, the use of decision tools that support Multi-criteria is very effective Hoceini and Abbas (2009b).

Our goal is to propose a new model of disambiguation based on a mathematical approach called MCDA. The basis of this method is to involve the collection of many criteria from various sources to form a mega rule that guides a parsing process. The advantage of this approach is to reduce the number of disambiguation scenarios discarding the dominated scenarios (i.e., scenarios with no better assessment and dominated by all used criteria) and classifying the effective scenarios (i.e., the ones that are not dominated) by a calculated overall score. All this is based on a clear definition of assessment criteria.

4.1. Main phases of Proposed Model

The establishment of a morphosyntactic disambiguation process based on multiple criteria decision requires us to follow a number of steps shown in Figure 3.

4.2. Description of the Approach

Our approach is summarized in the following steps:

- **Step 1:** *Compilation of a list of potential actions.*
  The establishment of a set of all possible solutions or actions. In our case, these solutions are the ambiguous tags. So, let A is the set $(a_1, a_2, \ldots, a_n)$, where a_i is considered like a candidate label, then a set of morphosyntactic information is generated.
- **Step 2:** *Constructing of a coherent family of criteria* $F = \{f1, f2, \ldots, fp\}$.
  Proper application of a multi-criteria approach requires a good choice for the applied criteria. These criteria are defined on the based of different concepts such as consistency, indifference, strict preference and comparability. However, developing a test that influences the choice of scenario i compared to another scenario is not an easy task.

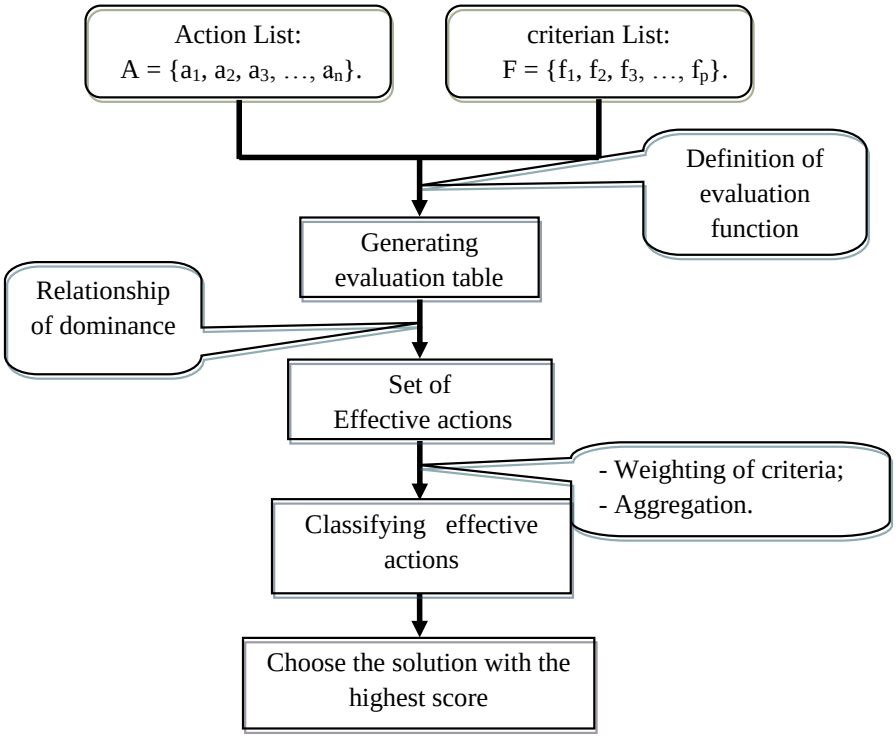


Figure 3. Different disambiguation techniques

But most importantly in defining a criterion is its power of discrimination between scenarios. In fact, discrimination becomes easier when the appropriate scenario is selected. However, a test that is discriminatory in some situations may not be so in other cases. Therefore, we need to construct a set of criteria that must meet three conditions namely: comprehensiveness, coherence and non-redundancy.

- **Step 3:** *Defining an evaluation function and an array of performance.*
  For each criterion we must generate an evaluation function that must be maximized or minimized depending on the type of the test used. The result of this function is a scorecard called the evaluation matrix. This later contains all the

evaluation results of each potential action when criteria are applied. Evaluation matrix rows correspond to the potential actions and the columns correspond to criteria. The matrix elements are the calculated estimates.

- **Step 4:** *Aggregation and criteria weighting*
  - **a) Aggregation:** it reduces the number of labels, and classifies them according to their overall scores. Choosing a method of aggregation will help standardize the evaluation table for better reading. To aggregate the different evaluations of a scenario calculated by the criteria, we propose to apply the TOPSIS aggregation method.
  - **b) Weighting:** it determines the weight of each criterion according to its importance[4]. So, weighting generates a vector of weights α, where each coordinate corresponds to a criterion. In our model, and to weigh the different criteria we adopt the Entropy weighting method.
- **Step 5:** *Selecting the label with the highest score*
  In order to obtain the scenario with the highest score, a classification of labels is performed decreasingly.

4.3. Aggregation Method : TOPSIS

4.3.1. Principle

The basis of the method is to choose a solution that is closest to the ideal solution, based on the relationship of dominance resulting from the distance to the ideal (the best on all criteria) and to leave the most of the worst possible solution (which degrades all criteria). TOPSIS is a multi-criteria method developed by Hwang and Yoon (1981). It reduces the number of disambiguation scenarios discarding the dominated ones, and ranking them according to their effective overall scores. In case of a tie, the closest scenario to the ideal, based on segregation measurements, is chosen.

4.3.2. Algorithm

- **Step 1:** Standardizing the performance (i.e., calculation of the normalized decision matrix); The normalized values e_{ij} are calculated as follows:

$$e'_{ij} = \frac{f_j(a_i)}{\sqrt{\sum_{i=1}^{m} [f_j(a_i)]}} \tag{1}$$

  With $i = 1, \ldots, m$, $j = 1, \ldots, n$, where $f_j\ (a_i)$ are the deterministic values of share(s) i for criterion j.

[4]The important criteria are able to discriminate between the solutions; and these criteria will have significant weights.

- **Step 2:** Calculation of the normalized and weighted decision matrix (i.e., calculating the product performance standard by the coefficients of relative importance of attributes). The matrix elements are calculated as follows:

$$e''_{ij} = \pi_j \cdot e'_{ij} \tag{2}$$

With $i = 1, \ldots, m$, $j = 1, \ldots, n$. π_j is the weight of j^{th} criterion.

- **Step 3:** Determination of ideal solutions (a^+) and anti-ideal solutions (a^-):

$$\begin{aligned} a^+ &= \{\max_i e"_{ij}, i = 1, \ldots, m; \ \& \ j = 1, \ldots, n\}; \quad e^*_j = \max_i\{e''_{ij}\} \\ a^+ &= \{e^*_j, j = 1, \ldots, n\} = \{e^*_1, e^*_2, \ldots, e^*_n\}; \\ a_- &= \{\min_i e"_{ij}, i = 1, \ldots, m; \ \& \ j = 1, \ldots, n\}; \quad e_{j*} = \min_i\{e''_{ij}\} \\ a_- &= \{e_j*, j = 1, \ldots, n\} = \{e_1*, e2*, \ldots, e_n*\}; \end{aligned} \tag{3}$$

- **Step 4:** Calculation of removal (i.e., calculate the Euclidean distance compared to the profiles a^+ and a^-). The distance between the alternatives is measured by Euclidean distance of dimension n. The remoteness of the alternative i with respect to the ideal (a^+) can be assimilated to the extent of exposure to risk and is given by:

$$D^*_i = \sqrt{\sum_{j=1}^{n} (e''_{ij} - e^*_j)^2} \tag{4}$$

$$D_{i*} = \sqrt{\sum_{j=1}^{n} (e''_{ij} - e_{j*})^2} \tag{5}$$

- **Step 5:** Calculating a coefficient that measures closeness to the ideal profile:

$$C^*_i = \frac{D_{i*}}{D^*_i + D_{i*}} \tag{6}$$

- **Step 6:** Storage of shares following their order of preferences (i.e., according to decreasing values of C^*_i; i is better than j if $C^*_i > C^*_j$).

4.4. Weighting Method : Entropy

4.4.1. Principle

The Entropy method is an objective technique for the weighting of criteria. The idea is that a criterion j is more important than the dispersion of stock valuations. Thus the most important criteria are those that discriminate most between actions (in our case actions are labels).

4.4.2. Algorithm

The entropy of a criterion j is calculated by the next formula Pomerol and Barba-Romero (1993):

$$E_j = -K \cdot \sum_{i=1}^{n} X_{ij} \cdot Log(X_{ij}).$$

where K is a constant chosen so that for all j, such as $0 \leq E_j \leq 1$, and $K = 1/\log(n)$ (n is the number of scenarios disambiguation). The entropy E_j is much larger than the values of e_j which are close. Thus, the weights are calculated according to the D_j (opposite of entropy):

$$D_j = 1 - E_j.$$

The weights are then normalized:

$$W_j = \frac{D_j}{\sum_j D_j}.$$

5. Implementation of the Proposed Solution

To better understand the proposed solution, we will keep the same approach mentioned above.

Let P ="رجع الوطنِ إلَى المغترِب", presented to our analyzer.

After segmenting the sentence into words, the analysis is done without any problem for units 2, 3 and 4. However, unit 1 "رجع" presents a typical morphological ambiguity. To remove this ambiguity we will apply our approach called multicriteria disambiguation as follows:

- **Step 1: Building a List of Analysis Scenarios:**
  The list (the set A) is obtained directly after the process of morphological analysis.

Verb	Scenario	Root
رجـع	فَـعَـلَ	رجـع
	فَـعِـلَ	رجـع
	فَـعُـلَ	رجـع
	فُـعِـلَ	رجـع

Table 1. Example of ambiguity generated when analyzing the verb "رجـع".

- **Step 2: Application of Criteria** To build a coherent family of criteria F, we propose two basic criteria to discriminate between the scenarios of the analysis: the test of vowel consistency, and the occurrence frequency test.

a) Criterion 1: Concordance of Vowels
This test will verify the correlation between the vowels of the lexical unit and the vowels of each candidate scenario. This test maximizes the function of assessment that goes with it is the addition (+).

a) Criterion 2: The Frequency of Occurrence.
This criterion is based on a statistical calculation on the basis of an annotated corpus so that the scenario that occurs most frequently will always score the highest. (Each appearance is one (1), so this is a test and to maximize the evaluation function that goes with it is the addition (+)). The results of applying this criterion are made on the basis of an annotated corpus is composed of 300 units spread over 10 arbitrarily selected paragraphs that are selected from (the books school school) an Algerian school textbook.

- **Step 3: Application of the Evaluation Function**
For both criteria (Concordance of vowels and frequency of appearance) the evaluation function is addition (+).
- **Step 4: Generating a Score Table (or score matrix)**

Scenario→ Criteria↓	S1"فَـعَـلَ"	S2"فَـعِـلَ "	S3"فَـعُـلَ"	S4"فُـعِـلَ"
Vowel Concordance	3	2	2	1
Appearance Frequency	16	5	2	1

Table 2. Evaluation Table (matrix).

- **Step 5: Aggregation and Weighting of Performance Criteria.**
Normalization of the scorecard is made by applying the formula (1) of the TOPSIS method.

Scenario→ Criteria↓	S1"فَـعَـلَ"	S2"فَـعِـلَ"	S3"فَـعُـلَ"	S4"فُـعِـلَ"
Vowel Concordance	0.71	0.47	0.47	0.24
Appearance Frequency	0.95	0.30	0.12	0.06

Table 3. Normalization of the Score Table.

a) Weighting of Criteria
In order to weight the criteria we use the entropy method, with respect of the initial condition mentioned in TOPSIS, i.e., the sum of the weights must be equal to 1. The following table shows the calculation Entropy values (E_j), the opposite of entropy (D_j) and normalization of weight (W_j) of the two criteria.

E_j	D_j	W_j
0.24	0.76	0.47
0.15	0.85	0.53

Table 4. Weighting the criteria

Note:
Checking the Status of weighting:

$$\sum_{j=1}^{p} W_j = W_1 + W_2 = 0.47 + 0.53 = 1.$$

(Condition tested).

b) Weighting of Evaluation Table (standard): This weighting is done using the formula (2) of the TOPSIS method.

Scenario→ Criteria↓	S1"فَعَلَ"	S2"فَعِلَ"	S3"فَعُلَ"	S4"فُعِلَ"
Vowel concordance	0.33	0.22	0.22	0.11
Frequency of appearance	0.50	0.16	0.06	0.03

Table 5. Weighting of Score Table

c) Calculation of Removal Measures
After applying formulas (3), (4) and (5), TOPSIS method reacts with different measures of distance for each scenario as illustrated in Table 6:

	S1"فَـعَـلَ"	S2"فَـعِـلَ"	S3"فَـعُـلَ"	S4"فُـعِـلَ"
D^*	0.33	0.22	0.22	0.11
D_*	0.50	0.16	0.06	0.03

Table 6. Weighting of Score Table

d) Calculation of the Measure of Closeness to Ideal Profile
To calculate coefficients C_i^*, we use the formula (6) of the TOPSIS method, and then establish a decreasing ranking of the factors. The scenario with the highest score is elected. So, these are the values obtained:

$$C_1^* = 1 > C_2^* = 0.32 > C_3^* = 0.24 > C_4^* = 0.$$

In our method the solution 1 "فَـعَـلَ" will be selected by the system, so the following morphological information will be generated.

	Information
Root	رجـع
Pattern	فَـعَـلَ
Tag	VAA3PMSIA
Designation in French	Verbe Accompli Actif 3e Personne Masculin Singulier Invariable Accusatif
Designation in English	Accomplished Verb Active 3rd Person Masculine Singular Invariable Accusative
Designation in Arabic	الفتح. على مبني الغائب، المذكر للمفرد للمعلوم مبني ماضي عل
Verb vowelized	رَجَـعَ

Table 7. Information generated by tagging the verb "رجع".

6. Conclusion

Using multiple criteria decision is a methodology that provides decision makers with tools to solve a decision making problem, taking into account several points of view. This paper attempts to present a new mathematical approach based on MCDA in order to categorize Multi scenarios of disambiguation and extract the best. This

method has the advantage of reducing dominated scenarios and ranking the rest by different evaluation criteria. Even though this technique is not widely used, it shows that the path of a multi-criteria analysis in NLP (based on recurrent common phenomena and to texts in all languages combined,) is very interesting. This technique offers an alternative and crucial complement method compared to systems that are based on a probabilistic approach and can be an indispensable complement to the model by contextual constraint.

Bibliography

Allotti, D. and C. Ponsard. Exposé sur l'étiqueteurs Statistiques et étiqueteurs par contraintes, 2005.

Aloulou, C., L. H. Belguith, A. H. Kacem, and A. Ben Hamadou. Conception et développement du système MASPAR d'analyse de l'arabe selon une approche agent. *RFIA*, 2004.

Chanod, J. P. and P. Tapanainen. Les étiqueteur statistiques et les étiqueteurs par contraintes, 1995.

Debili, F., H. Achour, and E. Souissi. La lague arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique. *IRMC*, 2002.

Hoceini, Y. and M. Abbas. Morphosyntactical Disambiguation Model of Arabic Based on a Multi-criteria Approach. In Arabnia, Hamid R. and David de la Fuenteand Jose A. Olivas, editors, *International Conference on Artificial Intelligence, ICAI 2009*, volume 2, pages 756–762, Las Vegas Nevada, USA, 2009a. CSREA Press.

Hoceini, Y. and M. Abbas. Une analyse multicritère de l'arabe. In *Journées d'étude du FSP France-Maghreb 'Pratiques langagière au Maghreb : corpus et applications*, Paris, France, Septembre 2009b. CERTAL - INALCO.

Hoceini, Y. and M. Abbas. Méthodologie Multicritère de Désambiguïsation Morphosyntaxique de la langue Arabe. In *3rd International Conference on Arabic Language Processing, CITALA'09*, pages 89–95, Rabat Morocco, May 2009c.

Hwang, C. R. and K. Yoon. *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag Berlin Heildelberg, New York, 1981.

Merialdo, B. Tagging english text with a probabilistic model. *Computational linguistics*, 1994.

Pomerol, J.C. and S. Barba-Romero. *Choix multicritère dans l'entreprise: principes et pratique*. Hermes, 1993.

Address for correspondence:
Youssef Hoceini
y_hoceini@yahoo.fr
Computer Science Institute,
Bechar University, P.B. 417,
Bechar 08000, Algeria

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 33–50

Machine Learning Approach for the Classification of Demonstrative Pronouns for Indirect Anaphora in Hindi News Items

Kamlesh Dutta[a], Saroj Kaushik[b], Nupur Prakash[c]

[a] National Institute of Technology, Hamirpur
[b] Indian Institute of Technology, Delhi
[c] Guru Gobind Singh Indra Prastha University

Abstract

In this paper, we present machine learning approach for the classification indirect anaphora in Hindi corpus. The direct anaphora is able to find the noun phrase antecedent within a sentence or across few sentences. On the other hand indirect anaphora does not have explicit referent in the discourse. We suggest looking for certain patterns following the indirect anaphor and marking demonstrative pronoun as directly or indirectly anaphoric accordingly. Our focus of study is pronouns without noun phrase antecedent. We analyzed 177 news items having 1334 sentences, 780 demonstrative pronouns of which 97 (12.44 %) were indirectly anaphoric. The experiment with machine learning approaches for the classification of these pronouns based on the semantic cue provided by the collocation patterns following the pronoun is also carried out.

1. Introduction

The automatic classification of indirect anaphora has attracted little attention of computational linguists. Indirect anaphora poses difficulty in designing anaphora resolution system required in various natural language applications (Mitkov, 1997) as the anaphor and antecedent do not exist explicitly in the text. Demonstrative pronouns have been found to be used as direct or indirect anaphora. For the purpose of the correct semantic interpretation of the text, it is important to be able to classify demonstrative pronouns as direct or indirect anaphora in the first instance and as-

Corresponding author: kd@nitham.ac.in
Cite as: Kamlesh Dutta, Saroj Kaushik, Nupur Prakash. Machine Learning Approach for the Classification of Demonstrative Pronouns for Indirect Anaphora in Hindi News Items. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 33–50. doi: 10.2478/v10108-011-0003-4.

sign correct semantic to the demonstrative pronouns acting as indirect anaphora in the next phase. Since explicit referent for indirect anaphora does not exist in the text, such an anaphora need to be identified and semantically understood in order to automatically understand the meaning of the text. This kind of anaphora is important for natural language tasks such as discourse resolution, information extraction, machine translation and language generation.

Among the recent activities in dealing with indirect anaphora (Fan et al., 2005) is based on Semantic path whereas (Gasperin and Viera, 2004) used word similarity lists for Portugeese corpus. Gundel et al. (2005) presented encoding scheme for indirect anaphora for Santa Barbara Corpus of Spoken American English. The work of Gundel et al. (2007) is based on the hypothesis of activation and focus hypothesis for New York Times news corpus. Kerstin and S.Hansen-Schirra (2003) presented multiplayer annotation for German News Paper corpus. Gelbukh and Sidorov (1999) presented indirect anaphora resolution based on the use of a dictionary of prototypic scenarios associated with each headword, and also of a thesaurus of the standard type. Boyad et al. (2005) have demonstrated the automatic classification of "it" for non-referential properties. Each work notes that dealing automatically with indirect anaphora is still a challenging task. All theories are based on semantic or conceptual structures and therefore automating their resolution requires more efforts. However one thing about the indirect anaphora is very clear that though it is inferable from the extended text, no explicit feature allow us to assign a relationship between anaphor and antecedent. Further the amount of such anaphora is sparse and a suitable automatic classification scheme needs to be evolved as its level of resolution does affect the anaphor resolution process.

In the present paper we develop an automatic classification scheme for indirect anaphora for Hindi text, which we believe, has not been attempted so far. Hindi has large number of demonstrative pronouns, which may have a direct referent or indirect one. We shall first identify the features that could be used for prediction of demonstrative pronoun's referentiallity. We shall also perform experiments using machine-learning algorithms to have an insight into the complexity of problem so that further refinements can be carried out. According to Schwarz (2001) we do not only categorize direct anaphoric relations, in which two expressions refer to the same extra-linguistic entity. In order to include more implicit relations between text elements, we also consider relations other than referential identity to be coreferential, which we call indirect anaphoric relations. A semantic and conceptual relation rather than a grammatical or lexical one links these identities. According to Mitkov (2002) indirect anaphora can be thought of as coreference between a word and an entity implicitly introduced in the text before. This gives rise to two problems with respect to the indirect anaphora: (a) detection of indirect anaphora, and (b) assigning an appropriate antecedent which in this case not available explicitly (Gelbukh and Sidorov, 1999).

2. Indirect Anaphora in Hindi

We first give a brief description of some key grammatical aspects of the demonstrative pronominal, and then discuss the issue of anaphoricity in Hindi. A list of possible demonstrative pronouns and their indirect anaphoricity behavior is given in Table 1. As evident, the number of pronoun usage is very large. Some of the pronouns can have indirect as well as direct anaphoricity whereas others have a direct antecedent in the discourse text.

The root form of these demonstrative pronouns is "yeh", "veh", "iss", "uss", "inn", "unn", "yahaan", "vahaan", "eissa", "veissa". The case marking modifies the pronouns and indicates the relation of pronoun with the neighbouring words. The case marker is added separately and the pronoun modifies accordingly. The agreement inflection is marked for person, number, and gender. In some readings the modified pronoun appears as a single word where as in others it is represented as two separated words. "inmein" "इनमे" (in these) can be written as "in mein" "इन में" or "inmein" "इनमें". Both forms are acceptable in written Hindi. However for our study we assume the modified pronoun as a single word. Various inflections after adding case marker to root word "iss" (this/it) is shown in Table 2.

Pronouns can appear as a noun or a modifier of noun. Noun form occurrences are governed by the case marking. Pronouns appearing as a noun in ergative, dative, and accusative forms require exact antecedent in the discourse. For example ergative cases (Pandharipande and Kachru, 1977), marked with case marker, "ne", expresses actor/ agent/ subject in perfective tenses for transitive verbs, as shown in sentence (1). The perfective form is indicative of pronoun + "ne" behaving as a noun phrase and the pronoun maps to some agent in the discourse. Non-animate nouns are not marked with ergative case. Therefore, normally the pronouns with these case forms do not exhibit the indirect anaphora.

(1) उन्होंने कहा कि महिला आरक्षण में विशिष्ट वर्गों के लिए अलग से आरक्षण की मांग सही नहीं है .
Unhon-ne kahaa ki mahilaa aarakshan mein vishisht vargon ke liye alag se aarakshan kii maang sahi nahiin hei.
He/She/They said that in the women's reservation demand for separate reservation for special category is not right.

On the other hand, several other forms of pronoun act as a modifier of noun and perfectly behave as a demonstrative pronoun. Such pronouns may be indirectly anaphoric as shown in sentence (2).

Pronoun in Hindi	Roman Gloss	English Pronoun	Indirect Anaphora
यह	yeh	this/it	yes
वह	veh	that	no
ये	ye	these	no
वे	ve	they	no
इस	iss	this/it	yes
इसे	isse	it	yes
इसी	isii	this	yes
उसी	usii	that	yes
इसका	isska	its	yes
इसकी	isskii	its	yes
इसके	isske	its	no
इसने	issne	it	no
इससे	iss-se	with it	no
इसमें	iss-mein	in it	yes
उस	uss	him/he/itr	no
उसे	usse	him/her/it	no
उसका	uss-ka	his/her/its	no
उसके	uss-ke	his/her/its	no
उसमें	uss-mein	in it	no
उसकी	uss-kii	his/her/its	no
उसने	uss-ne	he/she	no
उससे	uss-se	with him /her/it	no
उन	un	that/those	no
उन्होंने	unhon-ne	they	no
उन्हें	unhein	them	no
उनके	unke	by them, their	no
उनकी	unkii	their	no
उनका	unkaa	their	no
उनसे	un-se	them	no
उनमें	un-mein	in them	no
यहाँ	yhaan	here	no
वहां	vahaan	there	no
यहीं	yaheen	here	no
वहीं	vaheen	there	no
ऐसा	eissa	like this	yes
वैसा	vaissa	like that	no
ऐसी	eissii	like this	yes
वैसी	vaisii	lke that	no
ऐसे	eisse	like this	yes
वैसे	vaise	like that	no
इन	inn	this	yes
इनके	inke	about them	no
इनमें	inmein	in them	no
यही	yahii	this/it	no
वही	vahii	that	no

Table 1. Demonstrative Pronouns and its indirect anaphoricity

S.No.	Case	Pronoun Forms	Pronoun Hindi
1	Nominative Case	iss	इस
2	Ergative Case	iss-ne	इसने
3	Accusative Case	iss-ko	इसको
4	Instrumental Case	iss-se, isse iss-ke	इससे, इसे, इसके
5	Dative Case	is-ko, isse	इसको , इसे
6	Ablative Case	iss	इस
7	Genative Case	iss-ka, iss-ki, iss-ke	इसका, इसकी, इसके
8	Locative Case	iss-mein, iss-par	इसमें, इस पर

Table 2. Case marking of pronoun "iss"

(2) इस प्रकार उक्त निर्देश के आलोक में दोनों आरोपियों ने आज अदालत के समक्ष आत्मसमर्पण किया तथा ज़मानत याचिका दायर की थी .
Iss prakaar ukt nirdesh ke alok mein dono aaropion ne aaj adalat ke samaksh aatmsamarpan kiya tataa jamaanat yachikaa daayar kii thii.
Thus, in the light of the above directions both accused surrendered to the court today and filed bail petition.

The presence of words like "prakaar", "tarah", "baabat", after "iss" intuitively conveys that the pronoun is indirectly anaphoric and will not have a referent in the discourse. Further the presence or absence of case form or connective also helps us in assigning the indirect feature to our demonstrative pronoun as shown in sentence (3).

(3) इसी सिलसिले में पुलिस को दो महिलाओं की भी तलाश है
issii silsile mein police ko do mahilaon kii bhii talaash hei.
In this context police is in search of two ladies as well.

The presence of "mein" (in) after "silsile" (context) also conveys that the demonstrative pronoun "issii" (this) is a modifier and is adjunct to the sub sentence "police is in search of two ladies as well". The pattern "prakaar" if followed by auxiliary verb "hei (be) is directly referential. Therefore the role of connectives becomes important in the definition of referentiallity. Two cases in our text appeared in this form as shown in sentence (4).

(4) संहिता की प्रमुख विशेषताएं इस प्रकार हैं-
Sahinta kii pramukh visheshtayen iss prakaar hein.

Key features of Code are as follows:

Pronoun in a modifier can also have a direct referent in the discourse as shown in sentence (5).

(5) इस संस्थान के कार्यालय में नये छात्रों के स्वागतार्थ एक समारोह का आयोजन किया गया।
Iss sansthaan ke kaaryalya mein naye chaatron ke swaagatarth ek samaaroh kaa aayojan kiya gaya.
In the honour of new students a function was organized in the office of this institution.

The presence of noun "sansthaan" (institution) after "iss" is indicative of direct anaphoric feature of "iss".

Our approach is based on the occurrence of certain collocation patterns. We look at the collocation patterns occurring after demonstrative pronouns, if they do not have a nominal which may have appeared earlier, we see if it can be inferred as indirect anaphor by searching for occurrence of certain patterns. Some of commonly occurring patterns are "iss prakaar", "iss tarah", "eissii baat" etc. These patterns refer to a semantic category. Based on different information structures the pronouns are classified in different semantic categories and thus provide addition information that for these pronouns search for the antecedent should not be performed. Zaidan et al. (2007) also advocated the use of such additional information in the corpus.

We hypothesize that cognitive status of patterns following the demonstrative pronouns or personal pronouns account for the difference in the anaphoricity of the pronoun. Such patterns are known as collocation patterns. Common usage of collocation patterns along with pronouns and identifying their relationship, support "natural" choices of referent. Prasaad et al. (2004) used role of connectives in the development of Penn Discourse Tree Bank (PDTB) and (de Eugenio et al., 1997; Moser and Moore, 1995; Williams and Reiter, 2003) in Natural language generation. The findings reveal novel patterns regarding the collocation patterns for discourse and suggest additional experiments.

3. Methodology

The process of semantic classification of indirect anaphora required (a) selection of a corpus in Hindi, (b) identification of features that differentiate direct anaphora from the indirect one, (c) validation of our proposal using machine learning approach, and (d) development of automatic classification system for indirect anaphora. Our corpus should be encoded using Unicode. Hindi text using fonts which we may not be able to process seamlessly across different platform are not preferred. Identification of specific features requires careful analysis of corpus and formulation of appropriate rules. Since the data set is small, validation of scheme requires a selection of suitable algo-

rithms. In this paper we shall address first three issues. Development of automatic classification system will be carried out after fine tuning of our annotation scheme.

3.1. Corpus selection

We consider the data from Emille corpus. The corpus is based on the news items from Ranchi express (Sinha, 2002) and is the only known corpus in Hindi. The study aimed at improving the corpus with the semantic annotation for indirect anaphora. We analyzed 177 news items having 1334 sentences, 1600 demonstrative pronouns of which 97 (12.44 %) were indirectly anaphoric. The corpus is annotated for anaphora using scheme based on (Botley and McEnery, 2001) and customized for Hindi. Further Botley (2006) has also pointed out the limitation of his scheme and urged to encode more information essential for understanding indirect anaphora. This motivated us to further look into the annotation scheme adopted for the corpus.

Each occurrence of demonstrative pronoun is coded in an XML-compatible format so that it could be extracted automatically from the text. The indirect anaphora in this corpus is annotated as inferable antecedent. These are the cases that can be derived from the discourse but explicit noun phrase does not appear in the text. However existing encoding does not allows us to apply the resolution algorithms, as the exact antecedent cannot be extracted from the corpus. Further the pronoun marked as a direct or indirect, does not specifies what actually distinguishes direct anaphor from the indirect ones. We propose an extended scheme for annotating the corpus on indirect anaphora and incorporate features, which help us in identifying the indirect anaphoricity behavior of the pronoun. For our study we have considered only those pronouns, which have been marked as Inferable. The Emille corpus is based on the news items from Ranchi express and is the only known corpus in Hindi annotated for anaphora. The corpus is annotated for anaphora using scheme based on (Botley and McEnery, 2001) and customized for Hindi corpus by (Sinha, 2002). Each occurrence of demonstrative pronoun is coded in an XML-compatible format so that it could be extracted automatically from the text. The indirect anaphora in this corpus is annotated as inferable antecedent. These are the cases that can be derived from the discourse but explicit noun phrase does not appear in the text as a referent. The existing encoding does not allows us to apply the resolution algorithms, as the exact antecedent cannot be extracted from the corpus. Further, the pronoun marked as a direct or indirect, does not specifies what actually distinguishes direct anaphor from the indirect ones.

We propose an extended scheme for annotating the corpus on indirect anaphora and incorporate features, which help us in identifying the indirect anaphoricity behavior of the pronoun. For our study, we have considered only those pronouns, which have been marked as Inferable. The choice inspired by the work of Brown-Schmidt et al. (2005); Eckert and Strube (2000), these features captures preferences for NP- or non-NP-antecedents by considering a pronoun's predicative context. The underlying

assumption is that if certain pattern occurs after personal or demonstrative pronoun, then the pronoun will be likely to have a non-NP-antecedent.

3.2. Corpus annotation scheme

Theories proposed (Gundel et al., 2005) presents the case of indirect anaphora in English texts as a case of focus and attention. Kerstin and S.Hansen-Schirra (2003) have presented the scheme of annotating indirect anaphora. All these schemes were presented for English where it, that and this are generally used for demonstrative pronouns and also behaves as an indirect anaphora. (Dipper and Zinsmeister, 2009) annotated German corpus based on the semantic restriction and contextual features derived from the corpus. Navarretta and Olsen (2008) developed annotated Danish and Italian corpus for abstract anaphora.

Since indirect anaphora is based on cognitive kinds of relations, the classification may not be agreed upon between different annotators. However to start with we describe our own classification based on collocation pattern preference reflecting the key specific feature of our text corpus. The generalized classification proposed in (Fan et al., 2005) is based on abstraction, name-entity-relation, attribute relation and associative relation. However for Hindi corpus we adopt the classification scheme guided by the collocation pattern and the case marking that follows. The rationale of using this scheme is to keep the annotation process simple yet useful. As long as the annotator is spending the time to study example and classify it, it may not require much extra effort for classification.

The annotation scheme deals with the manual annotation of pronouns without an explicit noun phrase antecedent. Direct anaphors are able to find antecedent from noun phrases, the indirect anaphors are classified based on the semantic relations. The semantic classification ranges from explicit relations derivable from the information present in the discourse to implicit relations based on pure inference.

We focus once again on demonstrative pronouns and the ones marked as inferable in the corpus. We look at the collocation patterns for pronouns. The most popular approach for locating collocation patterns is the window-based which collects word co-occurrence statistics within the, context windows of an observing headword to identify word combinations with significant statistics-as collocations. For our experiment we have used the Heidelberg Tenka text concordance tool, an open source text analysis software and extracted the collocation patterns along with the pronouns as a head word and annotated the text as shown in Table 1. If the pronoun is indirectly inferable than pattern following the pronoun is also encoded and the semantic type is also specified according to the semantic classification given in Table 3. An example of annotation is shown in Example 6.

Feature	Value1	Value2	Value3	Value4	Value5
Distance Marking	P (proximal)	D (Distal)	None	None	None
Nature of deixis	P (Pronoun)	D (Demonstrative)	Z (Zero)	None	None
Recoverability of Antecedent	D (Directly Recoverable)	I (Indirectly Recoverable)	N (Non-recoverable)	0 (not applicable, e.g.) exophora)	None
Direction of reference	A (anaphoric)	C (cataphoric)	0 (not applicable, exophoric or deictic)	None	None
Phoric Type	R (Referential)	0 Not Applicable	None	None	None
Syntactic Function	M (Noun Modifier)	H (Noun Head)	0 (Not Applicable)	None	None
Antecedent Type	N (nominal)	P (propositional/ Factual)	C (Clausal)	J (Adjectival)	O (None)
Pronoun pattern	Pronoun and subsequent construct in the sentence				
Case marker/ Connective	Case marking or connective following the pronoun				
Semantic/ category	semantic categories as defined in Table 5				

Table 3. Feature Set used for annotation

Patterns following pronouns
samjhaa, aarakshan, liye, prakaar, baat, dishaa, sthiti, jaankaari, tarah, ek, paristhiti, roop, tak, kram, dhandhe, kuch, paksh, alaava, sandarbh, arth, or , gambhirta, siidhaa, tatvon, silsile, silsila, prashikshan, sambandh, gambhiirta, dushparinaam, kadam, galat, badii, dushparinam, ghatna, kaaranon, tamam, baavjood, saath , tayaari, matlab, manzar, moukaa, katthinaaii, baabat, sarvoch, saare_aaropon, suvidha, hii, baare, vyavasthaa, maukaa, maamla, sandesh, charchaa, aalok, suvidhaa, kitnii, prashnon, sambadh, sanchaalan, aashye, saath-saath, maansikta, durust, hinsak, gervajib, naaraz, koi, nai, vistrit, maamle, charchaaen, laabh, saari, saare, kaarnon , vishleshnon, seet, kuchh, khade, tahat, anapekshit, asar, ghatana, mudde, par, bhayaaveh, to, train, tayaarii, sab, siidha, tamaam, kathinaaion, baavzood, null
Case marker and connectives
mein, par, ki, kii, ke, se, hii, ka, ko, null, O
Semantic Categories
event, act, object, emphasize, subset, result, adjective, equivalence, type, summarize, reason, situation, context, additional, information, undefined

Table 4. Annotation feature set used for semantic annotation

(6) <s tag=2>झारखंड सरकार ने लातेहार, सिमडेगा, सरायेकेला और जामताड़ा को आज जिला बनाने संबंधी अधिसूचना जारी कर दी । </s><s tag =3> < w c= 1, tag="P,D,In,A, R,M,O, iss, prakaar, null, summarize"> इस </w> प्रकार अब झारखंड में जिलों की संख्या १८ से बढकर २२ हो गयी है । </s>
<s tag=18> राज्य में नए प्रशासनिक इकाईयों के गठन के सम्बन्ध में निर्णय लेने वाली उच्च स्तरीय समिति ने बैठक करके चार नये जिले बनाने की सिफारिश भी की थी । </s> <s tag=19> राज्य के मुख्य सचिव वी. एम. दुबे <w c=6, tag="P,D,D,A,R,M,N,iss, _, _,_"> इस </w> समिति के प्रमुख हैं । </s>

3.3. Classification

In most of the cases where pronoun is indirectly referenced the pattern following the pronoun is normally an abstract form of noun phrase, or characterization of the information conveyed in the discourse. This characterization cannot be capturing through the explicit referent, but a semantic annotation does provide the information about the status of information so far present in the discourse. A partial list of patterns and possible classification used in our experiment is listed in Table 4. In most of the cases "prakaar" is classified as "summarization" but if "prakaar" is followed by "ka/ki" then it is classified as "equivalence". Also in some cases two different annotators may classify same pattern differently. "iss-ke saath hii" (along with this only)

could be classified as an "event" and an "emphasize" as well. For our present study we include both the cases in our experiment.

Let
- S: list of tokens of semantic classification
- C: list of case markers and connectives {hii, ka, kii, ki, se, mein, par,…}
- T: list of tokens { "prakaar, "tarah", "kram",…}
- D: list of pronouns directly inferable but not indirectly inferable {issne, ussne, ussko, issko,…}
- R: list of remaining pronouns (these pronouns exhibit both type of behaviour) {yeh, iss, uss, inn,…}
- L: $D \cup R$
- SI: classification $SI \in S$
- XL: list of pronouns in the corpus
- X: current pronoun from the list XL; $X \in XL$
- XP: pattern following X
- XC: case marking
- ST: string consisting of X, XP, XC
- SN: syntactic category
- N: noun
- P: pronoun

For given pronoun X
1. Through concordance obtain string S which includes X, XP and XC
2. If $X \in D$ then skip to the next pronoun (pronouns defined purely for direct anaphora are eliminated from our study)
3. If a pronoun X is of noun type N and if the collocation pattern $XP \in T$ is an elaboration of one of the form from the classification list S then go to step 4
4. If a pronoun X is a modifier and if the collocation pattern XP following the pronoun X is an elaboration from one of the elements in classification list S, the pronoun is indirectly inferable.
5. If step 2 or step 3 is true then look for the connective/case marker $XC \in C$. If condition is satisfied annotate the given pronoun with X, XP, XC, SI along with other annotation provided in the Emille corpus else keep these features "null".

Classification rules
Since our classification scheme is based on the semantic cues provided by the concordance patterns of a discourse segment whose head is the pronoun with non NP-antecedent, we exploit this information for the purpose of classification. We have framed 25 rules, which can be applicable to a specific pronoun in a discourse. Some of the rules are given below:

Rule 1
IF : SN in H $\wedge$ PRONOUN in{iss} $\wedge$ XP in {prakaar} $\wedge$ XC in {null} $\Rightarrow$ CLASS = result

Rule 2
IF : SN in M $\wedge$ PRONOUN in {issii} $\wedge$ XP in {prakaar} $\wedge$ XC in {ka} $\Rightarrow$ CLASS = type

Rule 3
IF : SN in H $\wedge$ PRONOUN in {iss, issi} $\wedge$ XP in {tarah} $\wedge$ XC in {ke, ka} $\Rightarrow$ CLASS = type

Rule 4
IF : SN in M $\wedge$ PRONOUN in {iss, eisse} $\wedge$ XP in {tarah, tatvon, tamaam} $\wedge$ XC in {ki, kii, ke, ka, null} $\Rightarrow$ CLASS = type

Rule 5
IF : SN in M $\wedge$ PRONOUN in {ussii} $\wedge$ XP in {roop} $\wedge$ XC in {mein} $\Rightarrow$ CLASS = type

Rule 6
IF : SN in M, H $\wedge$ PRONOUN in {issii} $\wedge$ XP in {tarah} $\wedge$ XC in {null} $\Rightarrow$ CLASS = equivalence

Rule 7
IF : SN in M $\wedge$ PRONOUN in {issii, inn} $\wedge$ XP in {prakaar, saare} $\wedge$ XC in {se, null} $\Rightarrow$ CLASS = equivalence

Rule 8
IF : SN in M $\wedge$ PRONOUN in {ussii} $\wedge$ XP in {tayaarii} $\wedge$ XC in {ke} $\Rightarrow$ CLASS = adjective

Rule 9
IF : SN in M $\wedge$ PRONOUN in {inheen} $\wedge$ XP in {kaarnon} $\wedge$ XC in {se} $\Rightarrow$ CLASS = reason

Rule 10
IF : SN in M $\wedge$ PRONOUN in {issii} $\wedge$ XP in {paksh} $\wedge$ XC in {ki} $\Rightarrow$ CLASS = subset

Rule 11
IF : SN in M, H $\wedge$ PRONOUN in {yeh, iss, issii} $\wedge$ XP in {ek} $\wedge$ XC in {mein, ka, nom, null} $\Rightarrow$ CLASS = emphasize

Rule 12
IF : SN in M, H $\wedge$ PRONOUN in {yeh, iss, isse, issii, iss-ke, eisaa, eisse} $\wedge$ XP in {kram, gambhirta, silsile, silsila, ghatna, manzar, maamla, kuchh} $\wedge$ XC in {mein, ke, hii, ka, null} $\Rightarrow$ CLASS = event

Rule 13
IF : SN in M, H $\wedge$ PRONOUN in {iss, isse, isskii} $\wedge$ XP in {samjhaa, jaankaari, sambandh, baare, ghatana} $\wedge$ XC in {mein, kii, null} $\Rightarrow$ CLASS = information

When the pronoun has a direct NP-antecedent in the discourse the classification is categorized as direct only and pattern feature and case marker feature are not analyzed. The classification obtained suggests that the use of dictionary and thesaurus would improve the classification scheme.

Few examples of classifications based on the above rules are listed in Table 5.

Classification	Example
Event	जंगल बचाने का अभियान यहीं तक जारी नहीं रहा
Act	इस दिशा में चलाया जा रहा कार्य
Emphasize	यह एक सोची-समझी
People	इसी पक्ष की जांच-पड़ताल
Result	इसके लिए हमें मिलजुल कर कार्य करना होगा
Adjective	उसी तैयारी के साथ
Equivalence	इसी तरह की अन्य जातियां भी हैं
Type	इसी प्रकार का अधिकार
Summarize	इस प्रकार अब झारखण्ड में
Reason	इन्हीं कारणों से
Situation	ऐसी स्थिति का विरोध किया
Context	इन सन्दर्भ में
Additional	इसके बावजूद दूः स्थिति है कि
Information	इसकी जानकारी नहीं मिली

Table 5. Patterns and Classification for semantic annotation

3.4. Experiment

The distribution of anaphors with NP-antecedent (12.44 %) and non NP-antecedents (12.44 %) in Emille corpus is shown in Table 6. This figure is comparable to the number of pronouns without NP antecedents as reported in Gundel et al. (2005) as 16 % for New York times corpus, Poesio and Viera (1998) as 15 % or their corpus and Botley (2006) as 20 % for Associate Press corpus. All these studies are for English texts. We understand that this feature is similar across languages.

Though the present work deals with developing semantic annotation scheme for indirect anaphora in Hindi, the corpus obtained can be used for developing automatic classification models. (de Eugenio et al., 1997) has also applied the feature-based information in discourse for automatic generation of explanation in text generation. In our case the automatic classification of semantic categories can be used to automatically derive anaphora rules and ultimately use in anaphora resolution system. This will also prevent the human subjectivity, which is the main limiting factor in the de-

Pronouns	**direct**	**indirect**
yeh	184	11
iss	275	32
isse	23	2
issii	27	16
Iss-ka	18	1
isskii	15	1
issmein	12	1
usii	14	5
eisaa	29	2
eisee	13	11
eisse	23	4
yaheen	1	1
inn	47	1
inheen	2	1
Total	683	97
780	87.56 %	12.44 %
Total sentences: 1334		
Total demonstratives: 1600		

Table 6. Distribution of pronouns

velopment of large and reliable corpus. Two annotators may have different views about the category to which the given utterance should belong (Reiter and Sripada, 2002). We also experienced these problems in our attempts to tag the Emille corpus, which initially had some bugs, and our annotation was also based on our judgement, which cannot guaranty same results all time. This complexity of anaphor classification made us experiment with machine learning approaches.

After having tagged the data set it was easier for us to experiment with these methods. After trying several algorithms we chose to experiment with JRIP, J48 (the Weka implementation of C4.5) and LMT (Logical Model Tree)(Witten and Frank, 2005). First experiment included all the occurrences of demonstrative pronoun (with NP-antecedent and non NP-antecedents). Performance of J48 a C.45 decision tree based algorithm at confidence factor 0.8 improves to 88.462. Algorithm J48 computation time is far less than the LMT algorithm. Where J48 builds model in 0.02 seconds LMT algorithm 147.47 seconds. This makes J48 a preferred algorithm for very large

datasets. But since our corpus size is small, LMT gives a better model as it combines the advantage of regression and tree approach.

Data Set	JRIP			J48			LMT		
	S(%)	K	E	S(%)	K	E	S(%)	K	E
100	83	0.4684	0.0271	85	0.6488	0.147	84	0.6277	0.0310
200	86.57	0.6205	0.0227	88	0.7182	0.012	88	0.7213	0.0131
300	81.4545	0.4925	0.0293	86.5455	0.7073	0.0148	86.5455	0.7075	0.0978
400	82	0.4376	0.0277	86.5	0.6715	0.0143	85.75	0.6571	0.0155
500	85.7692	0.4202	0.0219	88.462	0.6598	0.0113	89.2308	0.6732	0.0116

E-Mean absolute error
S-Success Rate
K- Kappa Statistic

Table 7. Performance Measures of algorithms on given data sets

4. Analysis

The analysis of the experiment suggests that the performance measure in the current data set is dominated by the directly inferred pronouns. Experiment with the dataset excluding directly inferable pronouns resulted in a considerable drop in the performance in LMT from 89 % to 55 %. Performance of JRIP and J48 falls to 39 % and 42 % respectively. For reliable results, getting sufficiently large corpus is difficult. Further the linguistic cues used for the semantic classification of indirect anaphora needs further investigations as patterns like “prakaar”(10.31 %) and “tarha” (11.34 %) account for the major contribution toward the indirect referentiality of pronoun but other patterns like “tatvon”, “sthiti” and many others had marginal number of instances. Some patterns appeared only once. Other factor that we have ignored is the presence of words from other languages like English, which is becoming the natural way of communication and thus making the task of text processing more difficult.

The other solution could be the refinement of rules with usage of thesaurus in deciding the semantic classification, associating weight factor to positive classification and penalties for incorrect classification and specifying met rules. Further two annotators may also differ in their judgment about the class association. This would result in two different corpora for the same text. Also the annotator himself may not be able to decide exact category. In such cases either we may allow multi membership or assign different weights to the assignment. The possibility of inclusion of the indirect pronoun in different categories results in conflict in the present scheme. This conflict can be improved by incorporating a score value to each classification as follow: Premise of the rule $\Rightarrow$ { Class, likelihood} Where likelihood takes values as in the

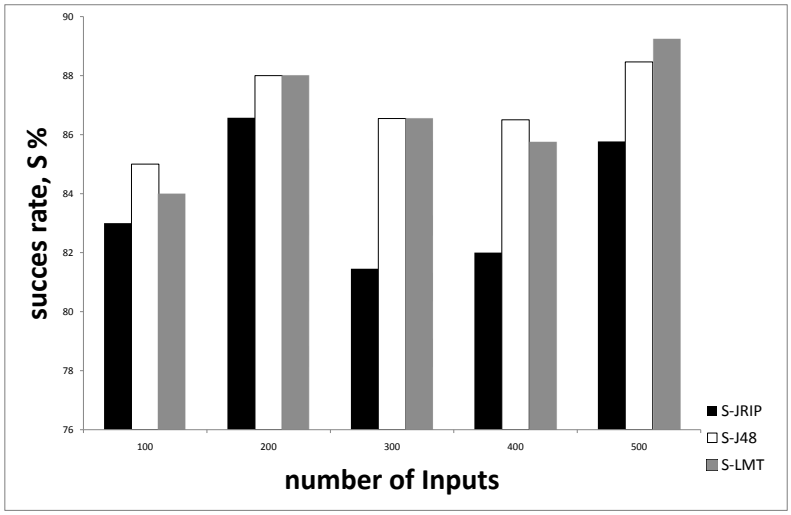


Figure 1. Success Rate of Algorithms on varied size of data sets

range of { -10 to +10 } ; positive value is for the likelihood of the correct classification, whereas negative values are indicative of the penalty of wrong classification.

Expanded rule specification could be Premise of the rule $\Rightarrow$ { ($Class_1$, $likelihood_1$), ($Class_2$, $likelihood_3$),..., ($Class_n$, $likelihood_n$) }.

Expanded rule can include the likelihood of class association for all classes. This requires more detail study of the corpus to decide upon exact likelihood values. In the present corpus the amount of instances available for indirect anaphora is too less to conclude concretely from the results obtained. Another possible solution is reduction in the number of classes by merging some of the categories. But in that case the extraction of semantic, which is useful in text cohesion, will be lost.

5. Conclusion

In this paper we have presented an enhanced annotation scheme on Emille corpus for indirect anaphora in Hindi. Annotation is enhanced with the semantic information for indirect anaphora. We experimented with automated classification using machine-learning approaches and our results show that the semantically enhanced annotation is a rich source of information for natural language understanding and

generation systems and for conducting data oriented research. Though the present model does not produce desirable results, fine-tuning of rules, incorporation more rules and with more data set better performance can be achieved.

Bibliography

Botley, S. and A. McEnery. Demonstratives in English: a corpus-based study. *Journal of English Linguistics*, 29:7–33, March 2001.

Botley, S. P. Indirect anaphora: Testing the limits of corpus-based linguistics. *International Journal of Corpus Linguistics*, 11(1):73–112, 2006.

Boyad, A., W. Geeg-Harison, and D. Byron. Identifying non-referential it: a machine learning approach incorporating linguistically motivated patterns. In *ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 40–47, Ann Arbor, June 2005. Association for Computational Linguistics.

Brown-Schmidt, S., D.K. Byron, and M.K. Tanenhaus. Beyond salience: Interpretation of personal and demonstrative pronouns. *Journal of Memory and Language 53 (2), pp. 292–313*, pages 292–313, 2005.

de Eugenio, B., J.D. Moore, , and M. Paolucci. Learning Features that Predict Cue Usage. In *ACL/EACL 97*, 1997.

Dipper, S. and H. Zinsmeister. Annotating Discourse Anaphora. In *Third Linguistic Annotation Workshop*, pages 166–169, Suntec, Singapore, August 2009. ACL-IJCNLP.

Eckert, M. and M. Strube. Dialogue acts, synchronizing units, and anaphora resolution. *Journal of Semantics 17 (1)*, pages 51–89, 2000.

Fan, J., K. Barker, and B. Porter. Indirect Anaphora Resolution as Semantic Path Search. *KCAP' 05*, October 2005.

Gasperin, C. and R. Viera. Using word similarity lists for resolving indirect anaphora. In *ACL Workshop on Reference Resolution and its Applications*, pages 40–46, Barcelona : Copisteria Miracle, S.A., 2004.

Gelbukh, A. and G. Sidorov. Word choice problem and anaphora resolution. *ISMT-CLIP*, 1999.

Gundel, J., N. Hedberg, and R. Zacharski. Pronouns without NP Antecedents: How do we know when a pronoun is referential. *Anaphora Processing: Linguistic, Cognitive and Computational Modelling, ed. by Antonio Branco, Tony McEnery, and Ruslan Mitkov. John Benjamins*, pages 351–364, 2005.

Gundel, J., N. Hedberg, and R. Zacharski. Directly and Indirectly Anaphoric Demonstrative and Personal Pronouns in Newspaper Articles. In *Proceedings of the Sixth Annual Discourse Anaphora and Anaphora Resolution Colloquium*, 2007.

Kerstin, K. and S.Hansen-Schirra. Coreference annotation of the tiger treebank. In *Workshop Treebanks and Linguistic Theories 200*, pages 221–224, 2003.

Mitkov, R. Factors in Anaphora Resolution: They are not the Only Things that Matter. A Case Study Based on Two Different Approaches. In *Proc. of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution*, 1997.

Mitkov, R. *Anaphora Resolution*. Longman, London, 2002.

Moser, M.G. and J. Moore. Investigating Cue Selection and Placement in Tutorial Discourse. In *ACL95*, 1995.

Navarretta, C. and S. Olsen. Annotating abstract pronominal anaphora in the DAD project. In *REC-2008*, May 2008.

Pandharipande, R. and Y. Kachru. Relational Grammar, Ergativity, and Hindi-Urdu. *Lingua*, 41:217–238, 1977.

Poesio, M. and R. Viera. A corpus-based investigation of definite description use. *Computational Linguistics*, pages 183–216, 1998.

Prasaad, R., E. Miltaski, A. Joshi, and B. Webber. Annotation and Data Mining of the Penn Discourse TreeBank. In *ACL Workshop on Discourse Annotation*, July 2004.

Reiter, E. and S. Sripada. Human Variation and Lexical Choice. *Computational Linguistics*, 28 (4):545–553, 2002. ISSN 0891-2017.

Schwarz, M. Establishing Coherence in Text. Conceptual Continuity and Text-world Models. *Logos and Language*, 2(1):15–24, 2001.

Sinha, S. A Corpus-based Account of Anaphor Resolution in Hindi. Master's thesis, University of Lancaster, UK, 2002.

Williams, S. and E. Reiter. A Corpus Analysis of Discourse Relations for Natural Language Generation. In *Corpus Linguistics*, 2003.

Witten, I. H. and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.

Zaidan, O., E. Jason, and C. Piatko. Using annotator rationales to improve machine learning for text categorization. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 260–267, Rochester, NY, April 2007.

Address for correspondence:
Kamlesh Dutta
kd@nitham.ac.in
National Institute of Technology
Hamirpur (HP)-177005, INDIA

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 51–62

Several Aspects of Machine-Driven Phrasing in Text-to-Speech Systems

Jan Romportl, Jindřich Matoušek

Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia

Abstract

The article discusses differences between a priori and a posteriori phrasing and their importance in the task of automatic prosodic phrasing in text-to-speech systems. On several examples it illustrates shortcomings of common evaluation of a priori phrasing performance using a posteriori phrasing of referential corpus data. The paper also proposes and evaluates a method for a priori phrasing based on template matching of quasi-syntactical representations of sentences.

1. Introduction

A very important prosody processing task in text-to-speech (TTS) systems is proper suprasegmental symbolic description of input sentences. Such a symbolic description can be called *prosodic structure* of a sentence. Knowledge of a prosodic structure of a synthesised sentence is vital for both explicit and implicit prosody generation techniques (by "explicit" we mean those techniques which explicitly produce surface prosodic features such as F0 or intensity contours, and by "implicit" we mean techniques where suprasegmental surface features emerge from concatenated segmental features, which is the case, for example, in unit selection TTS systems without signal modifications or often in HMM-based systems).

Our theoretical framework of prosody description (Romportl and Matoušek, 2005) understands prosodic structure in terms of relations among prosodic words, prosodic phrases, prosodic clauses, prosodemes and semantic accents. Especially prosodic phrases play an important role in naturalness of synthesised speech (Romportl, 2010a), and therefore prosodic phrase boundary estimation based purely on textual represen-

Corresponding author: rompi@kky.zcu.cz
Cite as: Jan Romportl, Jindřich Matoušek. Several Aspects of Machine-Driven Phrasing in Text-to-Speech Systems. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 51–62.
doi: 10.2478/v10108-011-0004-3.

tation of a synthesised sentence must be performed without major errors. It is one of the goals of this paper to propose and test a new algorithm which is able to designate prosodic phrase and clause boundaries in input TTS sentences so that the resulting phrasing is as much natural as possible (the question of prosodemes and semantic accents is left aside here).

Another goal, perhaps even more important, is to show that commonly used straightforward approaches to phrasing successfulness evaluation (i.e. comparison of automatically generated phrasing with referential testing data from a manually annotated corpus) are actually not very informative or fair because they ignore an essential fact about the nature of the prosodic phrasing problem.

2. Prosodic phrases

A prosodic clause is a continuous portion of speech between two pauses. It can comprise several prosodic phrases, and therefore prosodic phrases are often delimited by other prosodic features than a pause (e.g. intonation, segmental duration, etc.), thus their boundaries usually do not have special textual correlates such as punctuation marks.

A spoken utterance can usually be objectively segmented into prosodic phrases (Romportl, 2010a) because it already comprises relevant acoustic features actually produced by a particular speaker. However, in most cases it is not the only possible phrasing given the textual form of the utterance — the speaker could utter the text with different phrasing and it is also quite likely that if he utters the text once more, its phrasing will be different. This means that *a posteriori* phrasing of an utterance — i.e. the phrasing of an utterance already acoustically realised — is uniquely given, being a complex phenomenon determined by speaker's and listener's dispositions as well as by structural dispositions of the utterance itself. Both acoustic and syntactical features are important in the task of automatic *a posteriori* phrasing (Romportl, 2010b).

On the other hand, *a priori* phrasing of an utterance (or rather a sentence) is a process of purely *text-based* selection of one adequate phrasing from more potential variants which are allowed by the syntactical structure of the utterance. As a result of this, it is not correct to say that one particular *a priori* phrasing is correct whereas others are not: the sentence itself does not have enough causal potential to determine one particular phrasing

In the task of TTS synthesis we want to estimate the *a priori* phrasing of an input sentence, while this phrasing is then acoustically realised by the synthesis process itself. The question is, how to recognise whether the estimated phrasing is adequate for the given sentence or not. The immediate idea might be that we have a speech corpus of referential utterances with annotated phrases and we train and test the estimator using this corpus. However, this brings a serious problem: the *a priori* phrasing estimator is tested by the *a posteriori* phrasing annotations.

We can illustrate the situation by the following example. Lets suppose the speech corpus includes these two Czech utterances with annotated phrase boundaries (designated by "/"):

1a) Z mohutného kopce porostlého nízkými keři / se vine pěšina / do blízkého městečka.
(From mighty-Gen hill-Gen overgrown-Gen (by) low-Ins bushes-Ins / Refl winds footpath-Nom / to near-Gen town-Gen(-Diminutive).)

2a) Do sešlého hradu / zbořeného dlouhými věky / se vkrádá temnota / ze starého podzemí.
(To shabby-Gen castle-Gen / destroyed-Gen (by) long-Ins ages-Ins / Refl creeps in darkness-Nom / from old-Gen dungeons-Gen.

These two utterances have exactly the same syntactic structures, lexical words at the same positions bear identical morphological and syntactical categories (parts of speech, grammatical cases, syntactical functions), prosodic words at the same positions contain the same number of syllables, and still these two utterances have different *a posteriori* prosodic structures because 1a has three prosodic phrases whereas 2a has four. This means that there is not enough information in the textual form of an utterance to determine unambiguously its *a posteriori* phrasing. As a result, if a text-based phrasing estimator of a TTS system produces the *a priori* phrasing 2b, we really cannot say it is an error because there is no information available for the estimator to let it know that the "correct" phrasing form is 2a, not 1a.

2b) Do sešlého hradu zbořeného dlouhými roky / se vkrádá temnota / ze starého podzemí.

On the other hand, the *a priori* phrasing 2c can be considered as erroneous because it is in contradiction with the syntactic structure of the sentence (a tight syntactic relation between a noun "hradu/castle-Gen" and its attribute "sešlého/shabby-Gen" is disrupted by a phrase boundary):

2c) Do sešlého / hradu zbořeného dlouhými roky / se vkrádá temnota / ze starého podzemí.

Therefore, it is reasonable to impose requirements on a text-based *a priori* phrasing estimator so as the estimator avoids errors like 2c as much as possible while differences similar to the one between 1a and 2a (or 2a and 2b) do not matter.

It might seem that we are somehow trying to say what has been known for long: the placement of prosodic boundaries helps the listener parsing the sentence, hence they are highly correlated with syntactic boundaries, but to a large degree optional; however, at some places they would be rather confusing and this is considered wrong.

Such a statement is definitely true and well known, but this is not what we are aiming at here — instead, we are explicitly articulating the differences between *a posteriori* and *a priori* phrasing due to their influence on machine-learning and classification performance evaluation in the process of automatic *a priori* phrasing estimation.

A common machine-learning scheme would unnecessarily penalise the estimator's response 2b because the referential variant 2a is in the training/testing database. It would force the estimator to try to find some cues in the text of the sentence indicating that 2a is "correct" whereas 2b is not. But there are no such cues inherently present in the text — these cues might be found in speaker's dispositions, not in the sentence itself. And as the estimator does not have any access to what the speaker's dispositions can be, it will either continue to make these "false errors" (formally decreasing its nominal performance), or it will discover "false cues", which leads to overtraining.

A solution can be that there are all possible (or at least more) *a posteriori* phrasing variants of every sentence present in the corpus-based testing/held-out data for machine learning, allowing the machine learning algorithm to decide whether its output for a given feature-described sentence is correct (i.e. is one of the phrasing variants) or not. However, this is infeasible in normal situations when only one variant of each sentence is available, such as common speech corpora for TTS voices. Another solution, presented further in this paper, is more radical: it does not choose the approach of classical machine learning techniques or structurally driven construction of new prosodic structures for processed textual sentences; instead of this, it considers the whole TTS corpus (which is usually large) as the universe of all possible prosodic structures, and by a very simple algorithm it finds the most similar sentence to the processed one and reuses its phrasing.

By a machine learning technique we mean a process of automatic optimisation (usually iterative) of internal parameters of a classifier on the basis of training data (and possibly held-out data). The simple method proposed in this paper is a classifier, but its internal parameters are not optimised in any way, therefore no machine-learning technique is used.

Structurally driven construction of new prosodic structures refers to a process of building whole prosodic structures from smaller parts on the basis of various structural rules, such as those in grammar-based deterministic or stochastic parsing techniques. The proposed method does not use this approach as well — instead it takes prosodic structures already created in the corpus and does not consider any structural rules standing behind them.

3. Automatic a priori phrasing

As it was just mentioned, the idea behind our approach is following: if we have a suitable referential speech corpus (such as the one used as the source corpus for a given voice in a unit selection TTS system), we can understand all its utterances as templates and the phrasing estimation process is conceived as template matching

— an input TTS sentence receives the *a priori* prosodic structure (phrasing) which *a posteriori* belongs to the matched template sentence. This ensures that the selected assumed phrasing fits well with the syntactic structure of the given input sentence, and errors such as 2c are far less likely to occur than with other methods artificially constructing new phrasings which often might have not occurred in the corpus at all — e.g. HMMs, prosodic parsing, neural networks, etc., cf. (Romportl, 2010b; van Santen et al., 2008; Dutoit, 1997; Fitzpatrick and Bachenko, 1989).

3.1. Speech corpus

The template matching algorithm utilises a large collection of recorded utterances, which is usually not a problem in unit selection TTS systems where such data are necessary for speech segment database creation as well. It is even advisable to use the same corpus for both these tasks, because the unit selection algorithm will then process *a priori* phrasings originating from the same data as the concatenated segments.

For our experiments, we have used the corpus of 9,596 Czech declarative sentences recorded by a male speaker and used in the Czech TTS system ARTIC (Matoušek and Romportl, 2007). Prosodic phrases were automatically annotated in the whole corpus by a method based on artificial neural networks (Romportl, 2010b) trained on 250 manually inter-subjectively annotated sentences (Romportl, 2010a).

3.2. Syntactic features

A syntactic structure is a very important aspect in determining prosodic phrases. However, rather than the whole non-linear structure, it is more important for prosodic phrase boundaries to consider local syntactic relations between adjacent words, such as subject–attribute or predicate–object syntagmas (Palková, 1974). We proposed two sets of features for lexical word representation which proved suitable for automatic *a posteriori* phrasing (Romportl, 2010b):

- **Analytical functors** (`AFUN`). Analytical functors represent *syntactical functions* of lexical words. The inventory of functors we used originates from Prague Dependency Treebank 2.0. It has been slightly modified and it is listed in Table 1. Our whole corpus was syntactically parsed using the `TectoMT` application (Žabokrtský et al., 2008) with the McDonald's dependency parser yielding accuracy 85 % for Czech text. The parser assigns each lexical word an analytical functor, and since `AFUN` is a categorical feature, this functor is coded as a vector of 0's with a 1 in the dimension corresponding to the functor's order in Table 1 (e.g. *Obj* is coded as $[0, 0, 1, 0, 0, \ldots]$).
- **Apriori estimation of analytical functors** (`AFUNap`). Each lexical word form can be parameterised by a vector of *a priori* probabilities of analytical functions that this word form can appear in (e.g. $p(w = \mathrm{Pred}) = 0$, $p(w = \mathrm{Sb}) = 0.2$, $p(w = \mathrm{Obj}) = 0.5, \ldots$). The advantage of such a parameterisation is that no syntactical

abbrev.	description
Pred	Predicate
Sb	Subject
Obj	Object
Adv	Adverbial
Atv	Complement
Atr	Attribute
Pnom	Nominal predicate
AuxV	Auxiliary verb "be"
Coord	Coordination
Apos	Apposition
AuxTR	Reflexive tantum
AuxP	Preposition
AuxC	Conjunction
AuxOZ	Redundant or emotional item
AuxY	Adverbs and particles

Table 1. List of analytical functors.

parsing is needed — only a lexicon with word forms and probabilities which were derived from the data of Prague Dependency Treebank 2.0 in our case.

3.3. Template matching algorithm

1. Every sentence in the corpus is parameterised using the analytical functors of lexical words:
   (a) Each lexical word w_i of the sentence

   $$S_k : w_1, w_2, \ldots, w_p$$

   with p words is represented by a 15-dimensional feature vector $\mathbf{a}_i$ of `AFUN` or `AFUNap` (the choice between `AFUN` and `AFUNap` depends on the experiment; see the next section).
   (b) The parameterisation of the whole sentence S_k is given by the vector

   $$\mathbf{s}_k = \left[\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_p^T\right]^T. \quad (1)$$

   The vector $\mathbf{s}_k$ is thus an element of a 15p-dimensional space. The whole corpus creates as many spaces as there are different sentence lengths.
2. A sentence to be synthesised (further denoted as *input sentence*) has l lexical word tokens.

3. If $l < 5$, then the input sentence consists of a single prosodic phrase (and prosodic clause as well) and the algorithm ends. This is justified by the fact that there are only 31 sentences shorter than 5 words in the corpus, hence their phrasing variability can be omitted (there are only 6 phrasing variants for them anyway).
4. If $l > 9$, then prosodic clauses (and thus pauses) are determined in the input sentence in such a manner that each prosodic clause is a continuous part of the sentence between two adjacent punctuation marks (commas, hyphens, brackets, etc.). If a prosodic clause boundary is to be placed on a comma, the clause must be at least 4 lexical words long, otherwise the comma is inside the phrase and does not end it. The condition of 9 words is based on the fact that no prosodic phrase in the corpus was longer than 9 words.
5. If $l \leq 9$, then the input sentence is considered to be a single prosodic clause for now.
6. The whole input sentence is processed clause by clause. Each prosodic clause is further processed separately as if it were a standalone sentence.
7. The actually processed clause is l_C words long and is parameterised by a $15l_C$-dimensional vector $\mathbf{x}$ determined analogically to the steps (1a) and (1b) with the only difference that now it is a clause, not the whole sentence.
8. The sentence S_{k*} (the matched template) is found such that k^* is determined as

$$k^* = \arg\min_{k \in \mathcal{S}_{lC}} \|\mathbf{s}_k - \mathbf{x}\|, \tag{2}$$

   where $\mathcal{S}_{lC}$ is a set of indexes of those sentences from the corpus whose length equals to l_C.
9. Prosodic phrase boundaries in the actually processed prosodic clause are placed exactly as they are in S_{k*}.
10. If any of the phrase boundaries placed in (9) coincides with a punctuation mark not tagged as a clause boundary in (4), then this punctuation mark is newly considered to be a clause boundary (i.e. the actually processed clause can further be split into smaller clauses).
11. After processing all the clauses determined in (4) and (5), the phrasing of the whole input sentence is finished: the prosodic clause placement (and thus pause placement) is given by (4) and (10), the prosodic phrase placement inside of these clauses is given by (9).

Even though the syntactic and prosodic structures have a many-to-many mapping in the universe given by the corpus, the rule (8) ensures that only one prosodic structure is selected for the given input syntactic structure — the one belonging to the real corpus utterance with the closest syntactic structure to the input sentence.

The rules (3), (4) and (5) are clearly specific for this particular corpus, determined by the phrase length distribution in it. Sentences shorter that 5 words are omitted due to their low phrasing variability in the Czech language, sentences longer than 9 words are processed heuristically because there was no phrase longer than 9 words in the

corpus and we need a reasonable upper limit for the feature vector dimension. These values are presented here because they are probably more generally valid within the Czech language or at least the particular speaker, but there is technically no problem changing them in dependence on a corpus actually used.

4. Experimental evaluation

The algorithm described in the previous section is able to estimate *a priori* phrasing of any textual sentence so that this phrasing is consistent with the speaking style of the speaker who recorded the corpus. The key role is played by the formula 2 which expresses our hypothesis that the best *a priori* phrasing estimation of a so far unobserved sentence (or its part) is the *a posteriori* phrasing of an observed (in the corpus) sentence of the same length which is (quasi-)syntactically most similar to the unobserved sentence. This hypothesis can be justified by the following experiment using the collection of 4,824 sentences from the corpus whose length was 5–9 lexical words (this experiment excludes sentences longer than 9 words because the step (4) of the described algorithm is really just a heuristic rule technically allowing processing of longer sentences):

- The experiment is performed with the same number of iterations as the number of sentences in the collection (i.e. 4,824).
- In each iteration a tested sentence S_t is removed from the collection. From the rest of the collection, a sentence S_{k*} is selected according to the formula 2 for the sentence S_t. This is iteratively performed for all S_t from the collection.
- If in the particular iteration the referential phrasing of S_t is identical to the phrasing of S_{k*}, the counter of absolute agreement is increased by one.
- If the referential phrasing of S_t is not identical to the phrasing of S_{k*}, the difference is quantified as $\varepsilon = \|\mathbf{f}_t - \mathbf{f}_{k*}\|$. As S_t and S_{k*} are sentences p words long, the p-dimensional vector $\mathbf{f}_t$ represents the phrasing of S_t so that there are 1's in the vector at the positions corresponding to the indexes of the words at the phrase boundaries, and 0's elsewhere (e.g. for S_t: "word1 word2 / word3" $\mathbf{f}_t = [0, 1, 0]^T$. The vector $\mathbf{f}_{k*}$ analogically represents the phrasing of S_{k*}.

The experiment was performed separately for both `AFUN` and `AFUNap` parameterisations and the results are summarised in Table 2. It is clear that `AFUN` "outperforms" `AFUNap` in terms of the absolute agreement: in 26.1 % of the tested cases the *a priori* phrasing of the tested sentence was estimated identically to the referential *a posteriori* phrasing (the tested cases are whole sentences, not words). It might seem that this rate of absolute agreement is not high enough — but such a judgement would be a misinterpretation: we must bear in mind that we still test the *a priori* phrasing against the *a posteriori* phrasing. This value thus must not be simply interpreted as the *accuracy* in terms of a classification performance evaluation. It does not tell us much about the classifier we used (which is, anyway, trivial) — instead it tells us something more important about the data: only 26.1 % of the sentences in the corpus have their

	absolute agreement	$\mathrm{E}\{\varepsilon\}$
AFUN	1259 (26.1 %)	1.4111
AFUNap	888 (18.4 %)	1.4511

Table 2. Results of the experimental evaluation.

prosodic structures fully determined by their AFUN (quasi-)syntactic representations (and their linear distances).

Even though there are differences between referential and estimated phrasings in the remaining 73.9 % sentences from the collection, we can still assert that in spite of being different, an estimated phrasing is always a phrasing of a real utterance with a very similar syntactic structure (this is an analytical assertion), and therefore most likely fitting to the tested sentence (this assertion, though, should be corroborated by formal listening tests).

Moreover, the average value of ε shows that in those cases where the estimated *a priori* phrasing was not identical to the referential *a posteriori* phrasing, the average differences lie only in shifting one phrase boundary in each sentence. This interpretation of the average value $\mathrm{E}\{\varepsilon\}$ is based on the fact that $1{,}4111 \approx \sqrt{2}$ and if the vectors $\mathbf{f}_t$ and $\mathbf{f}_{k*}$ differ only in the placement of one element with the value 1 (e.g. $\mathbf{f}_t = [1,0,0,1,0]^T$ and $\mathbf{f}_{k*} = [0,1,0,1,0]^T$), then $\|\mathbf{f}_t - \mathbf{f}_{k*}\| = \sqrt{2}$. Of course this could also mean that there were 2 phrase boundaries added or deleted in every sentence, but after manual inspection of 100 randomly chosen tested sentences we verified that the most frequent difference really is a boundary shift, and most importantly, that the estimated *a priori* phrasing was always adequate for the given sentence, even though one boundary was shifted against the referential *a posteriori* phrasing — i.e. there were no errors similar to the example 2c, except for the cases where the real speaker recorded such inappropriate phrasing to the corpus (however, having the assumption that "the corpus is always right", these cases should not be considered as erroneous here — after all, the system tries to duplicate the speaking style of the original speaker as much as possible).

If we recalculate the values of the sentence absolute agreement and ε so that we consider the numbers of words in the sentences (i.e. the length distribution of the evaluated sentences, measured in lexical words), we get approximately 80 % accuracy of phrase boundary placement on words (including insertion and deletion errors). This accuracy value is fairly comparable with reports on English phrasing; no similar results allowing direct comparison have been reported for Czech. However, in our opinion it is not vital to further increase the word-level accuracy at any cost because our approach should guarantee that all the estimated phrase structures are appropriate in spite of possible phrase boundary insertions or deletions.

From the comparison of `AFUN` and `AFUNap` it is clear that it is better to have a syntactic parser as a part of the TTS system. However, if this is not possible for some reason, complete syntactic parsing can be replaced by the `AFUNap` approximation to some extent.

5. Conclusions

Our main goal was not to create a sophisticated algorithm for prosodic phrasing; rather we wanted to evoke more discussions on justness of many complicated machine-learning methods for prosodic phrasing by showing that even a very simple algorithm can efficiently fulfil this task once the apparent difference between *a priori* and *a posteriori* phrasing is considered as really constitutive for the view on the classification performance evaluation. Many common methods struggle for achieving higher accuracy in phrase boundary placement, forgetting that this often is — with a little hyperbole — rather a phantom chase. The most important thing is to clarify what we want: is it natural phrasing of synthetic speech, or is it the ability of the estimator to blindly follow its training/testing data? We have just wanted to point out that the former can be achieved by a simple algorithm based on the understanding that the corpus is all we know about prosodic phrasing and that if a new sentence comes, its *a priori* phrasing is same as the *a posteriori* phrasing of some sentence from the corpus. In our case, we have deliberately abandoned attempts to measure the phrasing successfulness in terms of the classification accuracy — instead we rely on a hypothesis that reusing of the phrasing of a real utterance syntactically similar to the processed one delivers an appropriate phrasing as well. The next step is to corroborate this hypothesis by large-scale formal listening tests following the scheme already used in the inter-subjective *a posteriori* phrasing annotation process of our corpus (Romportl, 2010a).

The algorithm proved well in the evaluation experiments and it can be easily implemented in a real TTS system. Its main advantages lie in its straightforward structure and its ability to generate adequate phrasing in almost all cases. The analytical functors used for parameterisation of words and sentences seem to be suitable as well. Still there are various aspects remaining unexplored: it might be interesting to see whether some optimisation of the algorithm parameters can improve its performance in terms of the absolute agreement — these parameters comprise mainly weights of particular functors in the formula for minimal distance of the sentence parameterisations. Since syntactic parsing is employed for analytical functor estimation anyway, it might also be helpful to utilise mutual syntactic relations of words in addition to their analytical functors, which would lead to more complex comparison and distance measuring. And finally the most important issue: the sentence template matching, as it is performed now, does not take into account rhythmical structure on the level of prosodic words; therefore features such as number of syllables or their distribution shall be added.

Acknowledgements

Support for this work was provided by the Ministry of Education of the Czech Republic, project LC536, and by the Grant Agency of the Czech Republic, project GAČR 102/09/0989. The access to the MetaCentrum computing facilities was supported by the research intent MSM6383917201.

Bibliography

Dutoit, Thierry. *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, Dordrecht–Boston–London, 1997.

Fitzpatrick, Eileen and Joan Bachenko. Parsing for prosody: what a text-to-speech system needs from syntax. In *In Proceedings of the Annual AI Systems in Government Conference*, pages 188 –194, Mar. 1989. doi: 10.1109/AISIG.1989.47324.

Matoušek, Jindřich and Jan Romportl. Recording and annotation of speech corpus for Czech unit selection speech synthesis. In *Proceedings of TSD 2007*, Lecture Notes in Artificial Intelligence, vol. 4629, pages 326–333. Springer, Berlin–Heidelberg, 2007.

Palková, Zdena. *Rytmická výstavba prozaického textu (Rhythmical Potential of Prose)*. Academia, Praha, 1974.

Romportl, Jan. On the objectivity of prosodic phrases. *The Phonetician*, 96:7–19, 2010a.

Romportl, Jan. Automatic prosodic phrase annotation in a corpus for speech synthesis. In *Proceedings of Speech Prosody 2010*, Chicago, IL, USA, 2010b.

Romportl, Jan and Jindřich Matoušek. Formal prosodic structures and their application in NLP. In *Proceedings of TSD 2005*, Lecture Notes in Artificial Intelligence, vol. 3658, pages 371–378. Springer, Berlin–Heidelberg, 2005.

van Santen, Jan, Taniya Mishra, and Esther Klabbers. Prosodic processing. In Benesty, Jacob, M. Mohan Sondhi, and Yiteng Huang, editors, *Springer Handbook of Speech Processing*, chapter 23, pages 471–487. Springer, Berlin, 2008.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, OH, USA, 2008.

Address for correspondence:
Jan Romportl
rompi@kky.zcu.cz
Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia
Univerzitní 8
306 14 Plzeň, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 63–76

Analyzing Error Types in English-Czech Machine Translation

Ondřej Bojar

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

This paper examines two techniques of manual evaluation that can be used to identify error types of individual machine translation systems. The first technique of "blind post-editing" is being used in WMT evaluation campaigns since 2009 and manually constructed data of this type are available for various language pairs. The second technique of explicit marking of errors has been used in the past as well.

We propose a method for interpreting blind post-editing data at a finer level and compare the results with explicit marking of errors. While the human annotation of either of the techniques is not exactly reproducible (relatively low agreement), both techniques lead to similar observations of differences of the systems. Specifically, we are able to suggest which errors in MT output are easy and hard to correct with no access to the source, a situation experienced by users who do not understand the source language.

1. Introduction

The Workshop on Statistical Machine Translation (WMT)[1] is a yearly open competition in machine translation (MT) among a few languages. Regularly, system outputs are manually judged using various techniques with the side-effect of establishing a trustworthy set of manual and automatic metrics (Callison-Burch et al., 2008, 2009). The manual evaluation methods tested so far are rather black-box, allowing to rank systems but revealing little or nothing about the types of errors in state-of-the-art MT.

A ranked list of error types of a system would be an invaluable resource for the developers of the system. In this paper, we use the WMT09 manual evaluation data

[1] `http://www.statmt.org/wmt06` to `wmt10`

 Corresponding author: `bojar@ufal.mff.cuni.cz`
Cite as: Ondřej Bojar. Analyzing Error Types in English-Czech Machine Translation . The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 63–76. doi: 10.2478/v10108-011-0005-2.

and our manual evaluation to identify error types in outputs of four English-to-Czech MT systems. Both techniques lead to similar results and we observe expectable but interesting differences in errors the systems make.

1.1. Techniques of Manual MT Evaluation

Traditionally, MT output has been manually judged by ranking of sentences in terms of adequacy and fluency. In WMT, the two axes of ranking were joined to a single one in 2008 due to a low inter-annotator agreement (Callison-Burch et al., 2008). Since 2009, WMT extends the sentence ranking with so-called "blind post-editing". The blind post-editing is performed by two separate persons in a row: the first one (the "editor") gets only the system output and is asked to produce a fluent sentence conveying the same message, the second one (the "judge") gets the edited sentence along with the source and the reference translation to confirm whether it is still an acceptable translation.

While the sentence ranking is hard to use for analysis of errors of individual systems, the blind post-editing provides a better chance. In Section 3, we design a simple technique for searching for MT errors given post-edits and apply it to four systems translating from English to Czech.

To support the observations, we also carry out an additional manual analysis: flagging of errors in MT output, see Section 4. This is a finer variant of post-editing and allows us to identify clear differences between types of MT systems in terms of errors they make. By linking the two types of manual evaluation, we are even able to observe that the systems differ in the possibility to correct particular error types in the blind post-editing task. Errors hard to fix in this setting are the most risky when the system is used by a user who does not understand the source language.

2. Brief Overview of Systems Examined

In the paper, we consider only a small subset of WMT09 systems. Still, they represent a wide range of technologies:

Google is a commercial statistical MT system trained on unspecified amounts and sources of parallel and monolingual texts.

PC Translator is a traditional commercial MT system tuned for years primarily for English-to-Czech translation.

TectoMT is an experimental system following the traditional analysis-transfer-synthesis scenario with the transfer implemented at the deep syntactic layer of language representation, based on the theory of Functional Generative Description (Sgall et al., 1986) as implemented in the Prague Dependency Treebank (Hajič et al., 2006). For the purposes of TectoMT, the tectogrammatical layer was further simplified (Žabokrtský et al., 2008; Bojar et al., 2009).

System	PC Translator	Google	CU-Bojar	TectoMT
Ranked $\geq$ others	**67%**	66%	61%	48%
Edits deemed acceptable	**32%**	**32%**	21%	19%
BLEU	**.14**	**.14**	**.14**	.07
NIST	4.34	4.96	**5.18**	4.17

Table 1. Manual and automatic scores of the four MT systems examined. Best results in bold.

CU-Bojar is an experimental phrase-based system the core of which is the Moses[2] decoder (Koehn et al., 2007). Considerable effort has been invested in tuning the system for English-to-Czech translation (Bojar et al., 2009).

Table 1 compares these systems on the WMT09 dataset using some of WMT09 evaluation metrics as reported in Callison-Burch et al. (2009). We see that TectoMT was distinctly worse than the other systems and that the two commercial systems perform better than the research ones. The traditional automatic metrics BLEU and NIST partially fail to predict this.

3. Exploiting Blind Post-Edits

As outlined above, the "blind post-editing" WMT dataset consists of source sentences, MT system outputs (also called hypotheses), edited outputs (also called edits) and yes/no acceptability judgments. Naturally, there is also the reference translation but its relation to the MT output is rather loose. Most of the relations in the dataset are one-to-many: There are always more MT systems for a single input sentence (each system provides a single best candidate), there are usually several manual edits of a given hypothesis and several judgment of a given edit.

The dataset is blind in several ways: the editor knows only the text of the hypothesis and neither the system, source text nor the reference translation. The annotator does not know the system or the editor either.

The edits are completely unrestricted and not formalized. All we have are two strings: the hypothesis and the edit. Editors are allowed to rewrite the sentence from scratch (but they usually don't have the capacity to do so because they don't know more than what is in the sentence).

3.1. Basic Statistics of the Dataset

The dataset consists of 100 source sentences. For the four systems in question, 29 unique editors provided the total of 1198 edits out of which only 708 (59%) contain a

[2] http://www.statmt.org/moses

new string.[3] Others were left unedited either because they were not comprehensible at all or because they were deemed correct. We are aware of the possible bias in our error analysis caused by ignoring esp. the incomprehensible sentences. The method discussed here is unfortunately not applicable to such cases, however the flagging of errors as described in Section 4 covers all the 100 sentences. In the sequel, we focus solely on the 708 edits.

The 708 edits were judged by 20 annotators, leading to the total of 2762 items (41% of which are marked as acceptable). In the sequel, we fully multiply the dataset so that an input sentence is duplicated as many times as any edit of any of the outputs was judged. This corresponds to micro-averaging all the observations over the dataset.

The average sentence length of a hypothesis is 21.4±9.8 words and the average sentence length of an edit is 20.6±9.3 words.

3.2. Generalizing Edits

In order to learn types of errors frequently done by individual MT systems, we need to somehow generalize the actual modifications performed in the edits. We use the following simple procedure:

1. Tokenize and morphologically analyze both the hypothesis and the edit.
2. Find differences between the two sequences of tokens. Various techniques can be applied here, we use the longest common subsequence algorithm (LCS, Hunt and McIlroy (1976)) as implemented in the Perl module `Algorithm::Diff` and the Unix `diff` tool. In future we would like to model block movements in the alignment as e.g. TER (Snover et al., 2009) or CDER (Leusch and Ney, 2008) do.
3. Synchronously traverse the tokens as aligned by the diff algorithm. Each step in the traversal is called a "hunk" and corresponds to an atomic edit.
4. Collect frequencies of seen types of hunks.

Figure 1 illustrates a hypothesis and an edit. There are four basic types of hunks, with the total frequencies given in Table 2: about 40k hunks link two identical tokens (Match)[4], 7k tokens were deleted from the hypothesis (Delete) and 5k were inserted (Insert). For about 12k tokens the LCS algorithms found sufficient context to mark them as being a substitute for each other (Modify). As we see in Table 2, individual edits vary a lot in terms of the number of these coarse hunk types. The edits that were approved in the second stage contain somewhat fewer matched tokens but the average sentence length for these edits is also slightly lower: 20.1±9.1. We would like to attribute this to a negative correlation between a hypothesis length and the acceptability of its edits (the percentage of judges who accepted the edit) but the correlation is rather weak: Pearson correlation coefficient of -0.13.

[3] One of the sentences had only the uninformative edits so we were left with 99 sentences.

[4] Actually, 1396 of these hunks have the same form but the morphological analyzer tagged them differently. We still count them as Match.

	Hunk	Hypothesis	Gloss	Edit	Gloss
1		Globální	Global	Globální	
2		finanční	finance	finanční	
3		krize	crisis.fem	krize	
4		je	is	je	
5		významně	notably	významně	
6	Modify	ovlivňoval	influenced.*masc*	ovlivňovala	influenced.*fem*
7		na	at	na	
8		akciových	stock	akciových	
9		trzích	markets	trzích	
10		,	,	,	
11		které	that	které	
12	Modify	se	*aux-refl*	prudce	quickly
13	Modify	pouštějí	send out	padají	fall
14	Delete	ostře	sharply	—	—
15		.	.	.	

Figure 1. Sample hypothesis and an edit, aligned using the LCS algorithm. Most of the hunks are "Match".

	Match	Delete	Insert	Modify
Total	39604	7176	4847	12261
Avg. per approved edit	13.4±6.6	2.5±2.6	1.8±1.9	4.2±3.2
Avg. per disapproved edit	15.0±7.0	2.6±2.9	1.7±2.0	4.6±3.3

Table 2. Coarse hunk types in the dataset of 99 input sentences with a valid edit.

3.3. Interpreting Hunks

As illustrated in Figure 1, the coarse hunk types do not always correspond to the change performed. The hunk 6 is an excellent example and we can directly derive the change from it. On the other hand, the hunks 12 to 14 are misaligned for our purposes. What actually happened was that the superfluous reflexive particle *se* got deleted, the lexical value of the verb got changed and the order of the adverb and the verb got swapped. For the purposes of this evaluation, we re-interpret only the Modify hunks handling the reflexive particle as a pair of Insert and Delete hunks.

Table 3 indicates how often a specific hunk class occurred in edits of an MT system output. We group hunks to the following classes:

Word matched if the form of the word is left unchanged (regardless a possible change in the automatically produced lemma or morphological tag).

Hunk Class	Count *% Approved*	CU-Bojar	TectoMT	Google	PC Translator
Word matched	39604	9781	7158	11176	11489
	38.5	*33.3*	*30.5*	*48.0*	*38.6*
Fix morphology only	2545	693	538	638	676
	33.6	*37.4*	*26.4*	*33.1*	*35.8*
Fix lexical choice, loose	1828	203	556	445	624
	39.5	*29.1*	*34.7*	*44.3*	*43.8*
Delete POS: N	1694	382	413	464	435
	39.1	*29.6*	*39.0*	*50.0*	*36.1*
Insert POS: N	1352	279	373	305	395
	41.8	*36.6*	*37.3*	*55.1*	*39.5*
Delete POS: V	1293	190	303	289	511
	40.8	*32.6*	*33.7*	*58.5*	*38.0*
Fix lexical choice, strict	1152	211	357	181	403
	37.8	*27.5*	*28.0*	*46.4*	*48.1*
Insert POS: V	990	199	179	212	400
	40.1	*38.2*	*33.5*	*51.9*	*37.8*
…					
Delete reflexive particle	437	97	132	110	98
	35.0	*23.7*	*17.4*	*61.8*	*39.8*
…					
Insert reflexive particle	385	41	67	99	178
	40.8	*24.4*	*29.9*	*52.5*	*42.1*
…					
Fix capitalization only	102	43	11	3	45
	31.4	*34.9*	*27.3*	*0.0*	*31.1*

Table 3. Most frequent hunk classes per system.

Fix capitalization only if the only difference between the word in the edit and the hypothesis is letter case.

Fix morphology only if the lemma of word is preserved but there is a change in the word form.

Fix lexical choice if the morphological tag is preserved but the lemma changes. We distinguish two subclasses: strict fix requires the exact same morphological tag[5] while loose fix requires only the identity of the part of speech.

Insert or delete reflexive particle if the Czech auxiliary particle *se* or *si* gets inserted or deleted. The particle is interesting because it is rather important for correct sense discrimination of some verbs but it is often placed at the second position in the sentence, possibly far away from the verb. In statistical MT systems, this

[5]This is an underestimate because the tagset sometimes uses a special value of a category indicating one of several possible simple values. The proper handling would thus be to unify the tags, not check them for identity.

particle gets often mis-aligned to some English auxiliary, e.g. *is*, and is spuriously produced in MT output.

Insert or delete words of various parts of speech, e.g. nouns (N) or verbs (V).

As we see in Table 3, the most frequent fix is related to pure change of morphology. This is a natural results because Czech has a very rich morphology and choosing the correct word form is the hardest part of English-to-Czech MT. In 33.6% of edits that included this type of fix, the second annotator approved the edit as a valid translation. Individual MT systems differ in the frequency this type of fix was applied: CU-Bojar and PC Translator needed a fix of the morphology most often. Google (thanks to its large n-gram language model) performed better in terms of necessary fixes but poorer in terms of acceptability of sentences with such a fix.

The fewest fixes of morphology were needed for TectoMT, a system that generates the target word forms using a deterministic morphological generator.

PC Translator seems to have the worst lexical choice (both strict and loose) followed by TectoMT. We are not surprised to see that CU-Bojar and Google need far fewer fixes of lexical choice as n-gram language models and longer phrases handle at least local lexical coherence well.

The acceptability judgments of edits with the following hunk classes are also noteworthy: fixing morphology in Google output is harder (leads to fewer edits accepted) than fixing lexical choice while quite the opposite holds for CU-Bojar. Again, we tend to attribute the difference to the language model size where it failed to guide CU-Bojar to the correct form and it misled Google to producing sequences output of bad words.

The reflexive particle was superfluously produced by TectoMT most often. Sentences with the superfluous particle were hard to correct (low acceptability rate) for TectoMT, where the sentence structure was probably distorted altogether, and easy to correct for Google, where the *se* was probably inserted as a mis-translation of an English auxiliary word.

Another frequent type of fixes is the insertion and deletion of nouns and verbs. We assume that a significant portion of these cases are word movements. Finally, we see that pure capitalization fixes are rare.

4. Flagging of Errors

To complement the manual judgments of WMT09, we carried out an additional manual evaluation of the four systems by marking errors in their output. We used an error classification inspired by Vilar et al. (2006), see Figure 2. Note that our annotators do not provide us with the full text of a corrected version of the hypothesis. Given our current experience, we believe that each of the annotators implicitly uses some "target acceptable output" and marks the changes necessary to reach it. Unlike in e.g. HTER (Snover et al., 2009), we have not recorded these target acceptable outputs in this exercise.

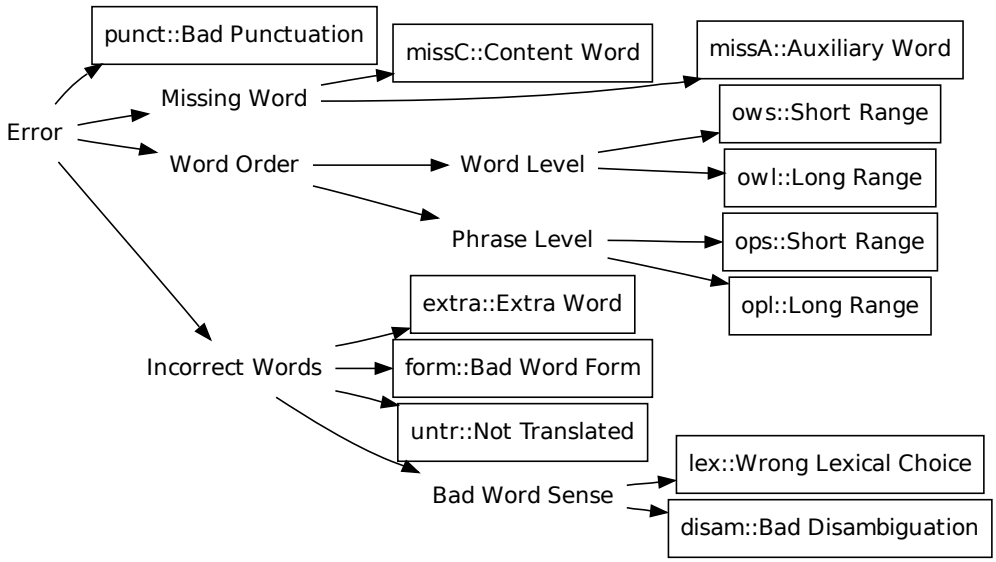


Figure 2. Error classification for manual flagging of errors. Boxes indicate the error flags used in our annotation.

Words appearing in the hypotheses can be marked as wrong for several reasons: they may not be translated despite they should be (untr), they may convey wrong meaning (Bad Word Sense; see below for details), they may be expressed in a bad morphological form (form) or they may be simply superfluous (extra). The annotators can add words that should have been in the hypothesis but they are missing (missC and missA). The set of allowed flags also covers some less important errors like punctuation or various types of word order issues. Short-range flags indicate that swapping a single unit with the next one would fix the problem, long-range flags indicate that the unit should be moved somewhere further away. If the misplaced words form a contiguous sequence ("phrase"), only one flag for the whole sequence should be used.

We used 200 sentences in total and 100 of them were the same sentences as annotated in the blind post-editing task. The annotation was carried out by 18 native Czech speakers to share the workload. Most of the sentences were annotated twice, 14% were annotated three times and 9% only once.

The instruction was to annotate as few errors as necessary to change the hypothesis to an acceptable output. An example of the annotation is given in Figure 3.[6] Unlike

Source	Perhaps there are better times ahead.
Reference	Možná se tedy blýská na lepší časy.
Gloss	*Perhaps it is flashing for better times.*
	Možná, že **extra::**tam jsou lepší **disam::**krát **lex::**dopředu.
	Perhaps, that there are better multiply to-front.
	Možná **extra::**tam jsou příhodnější časy vpředu.
	Perhaps there are favorable times in-front.
missC::v_budoucnu	Možná **form::**je lepší časy.
missC::in-future	*Perhaps is better times.*
	Možná jsou lepší časy **lex::**vpřed.
	Perhaps are better times to-front.

Figure 3. Flagging errors in outputs of four MT systems. English glosses are provided only for illustration purposes.

in the WMT09 blind post-editing, our annotators had access to the source and the reference. The identity of the MT system was hidden.

4.1. Agreement When Flagging Errors

The agreement when flagging tokens is relatively low. Excluding sentences with a single annotation, there were 5905 tokens flagged by at least one annotator. 43.6% of these tokens were flagged by all (two or three) annotators, regardless the number or type of error flags.

We attribute the low agreement to the fact that the annotators often diverge in the target acceptable output as well as in the set of marked corrections that lead to the target output. The agreement also drops if one of the annotators is willing to accept even slightly distorted output or forgets to mark some errors.

Table 4 provides the agreement for individual flag types on sentences with exactly two annotations. The highest agreement is achieved when labeling words not translated by the system but it is still surprisingly low. The flag `neg` was used by some annotators as a refinement of a bad form. We merge it with `form` annotations in other evaluations but we see that the agreement about negation is reasonable. The very low agreement in `case`, `opl` and `ops` is caused by only few annotators marking errors of this type.

We expected the `disam` and `lex` categories to be hard to distinguish. Disambiguation errors mean that the system has "misunderstood" the source word and picked a

[6] To avoid any systematic distortion of systems' outputs, our annotators were required to preserve the original space-delimited tokens. Several flags could have been assigned to a single token and this was often the case of tokens containing inappropriate punctuation, e.g. "I punct::form::doesn't, sleep." Some annotators also added special error marks for other minor errors such as letter case and bad tokenization. A few judgments also indicated that the sentence is totally wrong and not word marking individual errors (1 for PC Translator, 4 for Google and 6 for CU-Bojar and TectoMT).

	Flagged by				Flagged by		
Flag Type	One	Two	Agreement	Flag Type	One	Two	Agreement
untr	61	72	54.1	tok	24	4	14.3
neg	8	7	46.7	owl	116	17	12.8
extra	461	345	42.8	lex	559	63	10.1
form	1009	625	38.2	case	73	4	5.2
disam	912	310	25.4	opl	23	0	0
punct	304	98	24.4	ops	57	0	0
ows	258	69	21.1	Any	2614	2323	47.0

For each flag type we count tokens annotated by only one of two annotators and by both of them. Agreement = Two/(One + Two)

Table 4. Tokens flagged by one or two annotators.

clearly distinct wrong sense. All other (unexplained) bad lexical choices were marked `lex`. As we see, the agreement for `lex` is indeed very low. If we treat `lex` and `disam` as a single category, the agreement rises to 39.7%, more than the flag for erroneous word form.

In the following, we use all items that were flagged by any annotator. If a word is marked with the same flag by two annotators, we count it as two items.

4.2. Error Types by Individual MT Systems

Table 5 documents an important difference in error types made by individual systems. While CU-Bojar produced the fewest words with a bad sense (587), it missed by far the most content words (199). This is in line with the high score of the system in terms of NIST or BLEU and lower manual scores (see Table 1). Given the underlying technology, it also suggests a certain overfitting in the tuning of the underlying log-linear model, e.g. the penalty for producing a word set too high. On the other end of the scale is PC Translator which had the fewest content words missing (42) but did not score particularly well in terms of lexical choice (800). Google seems to choose a good balance (72 missed content words, 670 wrong lexical choices).

We also see that systems with n-gram LMs perform better for some less serious phenomena like local word order (ows) and punctuation (punct).

Finally note that the overall number of errors or serious errors marked by humans does not correlate with other manual evaluations (Table 1). The number of errors marked in PC Translator's output, the best ranked system, was higher than e.g. Google. Admittedly, the set of flagged sentences is not the same but still it comes from exactly the same test set of WMT09 and covers the blind post-editing subset. This again indicates, how difficult the evaluation of MT is even for humans.

	Google	CU-Bojar	PC Translator	TectoMT	Total
disam	406	379	569	659	2013
lex	211	208	231	340	990
Total bad word sense	617	587	800	999	3003
missA	84	111	96	138	429
missC	72	199	42	108	421
Total missed words	156	310	138	246	850
form	783	735	762	713	2993
extra	381	313	353	394	1441
untr	51	53	56	97	257
Total serious errors	1988	1998	2109	2449	8544
ows	117	100	157	155	529
punct	115	117	150	192	574
owl	43	57	50	44	194
ops	26	14	25	15	80
letter case	13	45	24	21	103
opl	10	11	11	13	45
tokenization	7	12	10	6	35
Total errors	2319	2354	2536	2895	10104

Table 5. Flagged errors by type and system.

4.3. Errors Easy and Hard to Fix in Blind Post-Editing

Table 6 indicates which errors of a particular system are easy to fix in blind post-editing and which are particularly hard. The higher the number, the easier to fix errors of that kind. We obtained the scores as the difference in error distributions in top and bottom 25% of sentences when sorted by the average acceptability of post-edits of the sentence.[7] For instance, 30.30% of errors made by Google in 25% most easily post-editable sentences were errors in form. The percentage of errors in form rises to 32.90% if we look at 25% sentences that were hardest to post-edit. Table 6 shows the difference of these figures, indicating that errors in form by Google are relatively hard to fix (-2.60) in blind post-editing.

This kind of evaluation confirms our expectations about similarities and differences of the examined MT systems and it is in accordance with the post-edits alone, see Section 3.3: lexical choice is a problem hard to fix for every system. Although the "lex" category is very similar to "disam", they were probably easy to distinguish in the output of TectoMT: we know that TectoMT's dictionary is not clean and often

[7] As we know from previous section, each edit was judged by several judges. We denote the percentage of approvals as the "acceptability" of an edit and average those numbers over all edits of a hypothesis. Note that the order of sentences by the average acceptability of its post-edits is different for each system.

System	Easy to Fix	Hard to Fix
CU-Bojar	form (11.0), tok (3.3), punct (2.9)	disam (-4.0), extra (-4.9), lex (-5.8)
TectoMT	missA (4.4), disam (4.2), ows (2.2)	untr (-1.6), missC (-2.3), lex (-7.3)
Google	missA (6.6), punct (6.1), ows (3.5)	form (-2.6), missC (-2.9), lex (-8.3)
PC Translator	ows (7.3), punct (5.3), missA (2.1)	disam (-2.7), extra (-7.7), lex (-7.9)

Table 6. Errors easy and hard to fix in blind post-editing.

suggests a rather weird lexical choice, no language model is applied to disambiguate better. This is confirmed in our table: such clear disambiguation flaws were easy to fix even without access to the source sentence because most post-editors speak English and could guess what the original word was.

The interesting difference between Google and CU-Bojar, both using phrase-based translation and n-gram language model, mentioned in Section 3.3 is more pronounced here. While errors in form in CU-Bojar's output are easy to fix (11.0), they are rather hard to fix in Google's output (-2.6). We attribute the difference to the strength of Google's language model: errors in form include errors in negation and the overall more or less fluent output can easily mislead post-editors. CU-Bojar uses a smaller language model and the errors in form probably cause output more incoherent than deceiving. Similarly, errors in form are not among the most serious problems in PC Translator output. While other systems confuse post-editors by missing content words (missC), PC Translator tends to confuse them by additional words (extra).

5. Conclusion

This paper attempted to reveal and quantify differences between error types various MT systems make when translating from English to Czech. The first dataset used consisted of the WMT09 blind post-edits. To complement this type of evaluation, we manually marked errors in the same set of system outputs.

Both types of manual evaluation can be used to reveal more about individual MT systems. While the reproducibility of each of the evaluations is relatively low (annotators diverge in errors they mark or post-edit), the overall picture provided by both evaluation types is rather similar: Statistical systems were somewhat better in lexical choice (probably thanks to the language model) while the fewest morphological errors can be achieved either by a large language model or a deterministic morphological generator. The drawback of a powerful language model is the risk of misleading: a fluent output is not a good translation of the source text.

We have suggested a method for detailed analysis of blind post-editing data. Given the availability of this manually created resource for various language pairs at WMT evaluation campaigns, we hope researchers will be able to focus on most serious errors of their specific MT systems.

Acknowledgement

The work on this project was supported by the grants P406/10/P259, P406/11/1499, and the project EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic).

We are grateful to all our student annotators and also to the anonymous reviewers for their comments on previous versions of the paper.

Bibliography

Bojar, Ondřej, David Mareček, Václav Novák, Martin Popel, Jan Ptáček, Jan Rouš, and Zdeněk Žabokrtský. English-Czech MT in 2008. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March 2009. Association for Computational Linguistics.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0309`.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Josh Schroeder. Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March 2009. Association for Computational Linguistics.

Hajič, Jan, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4, 2006.

Hunt, James W. and M. Douglas McIlroy. An Algorithm for Differential File Comparison. Computing Science Technical Report 41, Bell Laboratories, June 1976.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-2045`.

Leusch, Gregor and Hermann Ney. BLEUSP, INVWER, CDER: Three improved MT evaluation measures. In *NIST Metrics for Machine Translation Challenge*, Waikiki, Honolulu, Hawaii, Oct. 2008.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands, 1986.

Snover, Matthew, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric.

In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1626431.1626480.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. Error Analysis of Machine Translation Output. In *International Conference on Language Resources and Evaluation*, pages 697–702, Genoa, Italy, May 2006.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA, 2008.

Address for correspondence:
Ondřej Bojar
bojar@ufal.mff.cuni.cz
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25
11800 Praha, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 77–86

Quiz-Based Evaluation of Machine Translation

Jan Berka[a b], Martin Černý[a], Ondřej Bojar[b]

[a] Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering
[b] Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

This paper proposes a new method of manual evaluation for statistical machine translation, the so-called quiz-based evaluation, estimating whether people are able to extract information from machine-translated texts reliably. We apply the method to two commercial and two experimental MT systems that participated in WMT 2010 in English-to-Czech translation. We report inter-annotator agreement for the evaluation as well as the outcomes of the individual systems. The quiz-based evaluation suggests rather different ranking of the systems compared to the WMT 2010 manual and automatic metrics. We also see that overall, MT quality is becoming acceptable for obtaining information from the text: about 80% of questions can be answered correctly given only machine-translated text.

1. Introduction

There are many ways for evaluating the quality of machine translation, from automatic metrics like BLEU (Papineni et al., 2002) or METEOR (Lavie and Denkowski, 2009), to different kinds of human judgement (manual evaluation) (Callison-Burch et al., 2010).

These methods are based on the question "Is this a plausible translation of the text?" We propose a different manual evaluation method, which asks a slightly different question: "Does the translated text provide all the information of the original?" This follows the idea, that in many real-life situations like reading the news or getting travel directions we do not need to have a totally correct translation—we just need to now what happened or where to go.

Our proposed quiz-based evaluation method is centered around yes/no questions. We start by collecting naturally occurring text snippets in English, manually equip

Corresponding author: bojar@ufal.mff.cuni.cz
Cite as: Jan Berka, Martin Černý, Ondřej Bojar. Quiz-Based Evaluation of Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 77–86. doi: 10.2478/v10108-011-0006-1.

them with a set of yes/no questions (in Czech) and translate them using four MT systems to Czech. The translated texts are then handed to annotators, who see only one of the machine translations and answer the questions. We measure the quality of translation by the number of correctly answered questions.

2. Preparation of Texts and Questions

For the experiment, we collected English texts from various sources, written hopefully by native speakers.[1] These texts covered four topic domains:

- Directions description – these texts provided information of a location of a certain place, or described a route to somewhere,
- News – this topic contained snippets from newspaper articles about politics and economy,
- Meeting – texts from this domain contained information about places, times and subjects of different meetings,
- Quizes – short quiz-like questions in the fields of mathematics, physics or software engineering.

These topics cover a large variety of common texts, from which the reader needs usually only the core information. The grammatical correctness of the MT output is not important as long as the meaning is not disrupted.

The collected texts had also three different lengths from one sentence texts to texts with two and three sentences. This distribution of texts allowed us to examine, whether some topics are harder to translate and if success of the translation (from the point of view of quiz-based evaluation) depends on text length.

We managed to collect a total of 132 texts with close to uniform distribution of topic domains and lengths.

In the next step, we created three questions with answers yes or no for each text. This meant the total of 396 different questions for evaluation of the machine translation systems. Figure 1 shows four single-sentence sample texts and the corresponding questions (in Czech with an English gloss).

After the texts were collected and questions were prepared, we did a final preannotation check of the "golden" answers (answers deemed correct by authors of the questions). In this process, 78 answers were changed, 12 of them with no change of the actual value and changing only the uncertainty indicator (in situations when it was natural and right for an annotator to be unsure). 8 questions were completely removed. We ended up with 376 questions with the following distribution of golden answers: 191 yes, 170 no, 15 can't tell.

Texts were then translated by four different machine translation systems (see Section 3). Each annotator was given a set of 132 texts with the corresponding ques-

[1] We always chose web sited in countries where English is the official and majority language. In the current globalized world, the mother tongue of the author can be different.

Topics	Texts and questions
Directions	Follow the red arrows to the registration desk.
	Jsou šipky zelené? Are the arrows green? Ukáže cestu asistent? Will an assistent show you the way to registration desk? Does the registration take place right by the entrance? Probíhá registrace hned u vchodu?
News	The Chinese government is considering legislation that would make eating cats and dogs illegal.
	Je v Číně zakázáno jíst psy? Is dog eating banned in China? Uvažuje čínská vláda o zákazu pojídání psů a koček? Is government considering a ban of dog and cat eating? Jí v Číně psi často kočky? Do dogs in China often eat cats?
Meetings	The University of York Filmmaking Society meets every Monday at 6.30pm at L/047.
	Existuje na univerzitě v Yorku spolek filmařů? Does a filmmaking society exist on University of York? Je v Yorku zřejmě filmová univerzita? Is a film university in York? Konají se schůzky každé pondělí? Do the meetings take place every monday?
Quiz	A equals two thirds and B equals free fifths.
	Je A větší než B? Is A greater than B? Jsou A a B stejně velké? Does A equal B? Je B menší než 1? Is B less than 1?

Figure 1. Examples of one-sentence texts and their corresponding questions

tions. We tried to get annotations of all topics, lengths and MT systems uniformly distributed, but not every annotator completed the task. In total we obtained a set of 1891 annotated texts, with the distribution of topics and lengths as shown in Table 1 and MT systems as shown in Table 2.

The use of yes/no questions slightly affected the possibilities of questioning, but allowed us to process the answers automatically. The annotators were given 6 possible answers to choose from:

- yes, denoted by annotation mark 'y',
- probably yes (marked as 'Y'),
- no ('n'),
- probably no ('N'),
- can't tell (based on the text), marked as 'x',
- don't understand the question ('X').

Except for 'X', the capital letter was used to indicate that the annotator was not sure.

	1 sentence	2 sentences	3 sentences	All lengths
Directions	10.4%	8.2%	8.5%	27.1%
Meetings	7.0%	6.1%	7.1%	20.1%
News	10.3%	10.0%	8.8%	29.1%
Quizes	8.5%	9.6%	5.6%	23.7%
All topics	36.2%	33.9%	30.0%	

Table 1. Topic domains and lengths distribution in annotated texts

	Sentences	Google	CU-Bojar	PCTrans	Tectomt
Directions	1	23.2%	25.8%	29.0%	22.1%
	2	25.0%	23.7%	27.0%	24.3%
	3	29.3%	18.5%	23.6%	28.7%
Meetings	1	24.5%	32.0%	21.1%	22.5%
	2	23.0%	24.8%	23.9%	28.3%
	3	26.0%	22.7%	30.7%	20.7%
News	1	23.6%	25.7%	26.7%	24.1%
	2	24.2%	30.8%	23.6%	21.4%
	3	23.0%	26.7%	25.5%	24.9%
Quizes	1	25.3%	18.5%	26.5%	29.6%
	2	27.4%	21.2%	20.7%	30.7%
	3	24.3%	27.2%	22.3%	26.2%

Table 2. MT systems distribution in annotated texts (with respect to topic domains and text lengths)

3. Brief Overview of Examined Systems

In this paper, we consider 4 systems from WMT10. It is a small subset of all the systems present, but they represent a wide range of technologies.

Google Translate is a commercial statistical MT system trained on unspecified amount of parallel and monolingual texts.

PC Translator is a Czech commercial system developed primarily for English-to-Czech translation.

TectoMT is a system following the analysis-transfer-synthesis scenario with the transfer implemented at a deep syntactic layer, based on the theory of Functional Generative Description (Sgall et al., 1986) as implemented in the Prague Dependency Treebank (Hajič et al., 2006). For TectoMT, the tectogrammatical layer was further simplified (Žabokrtský et al., 2008). We use the WMT10 version of TectoMT (Žabokrtský et al., 2010).

CU-Bojar is an experimental phrase-based system based on Moses[2] (Koehn et al., 2007), tuned for English-to-Czech translation (Bojar and Kos, 2010).

4. Results

4.1. Intra-annotator Agreement

In order to estimate intra-annotator agreement, some texts and the corresponding questions in the set of 132 texts given to each annotator were duplicated. The annotators were volunteers with no benefit from consistent results, so we didn't worry they would search their previous answers to answer repeated questions identically. In fact, they even didn't know that they have identical texts in their set.

However, the voluntary character of the annotation has also caused troubles, because we got only very few data for the intra-annotator agreement. Only 4 annotators answered questions about two identical texts, with the average intra-annotator agreement of 92%.

About two months after the annotation, one of the annotators answered once again all the questions from his set of texts, providing a dataset of 393 answered questions. From the comparison of his new and old answers we estimate the intra-annotator agreement as 78.9%.

4.2. Inter-annotator Agreement

In order to estimate the inter-annotator agreement, each translated text with corresponding questions was present in several sets given to independent annotators. The inter-annotator agreement between two annotators x, y was then computed as:

$$\mathrm{IAA}(x,y) = \frac{\text{number of identically answered questions}}{\text{number of common questions}} \tag{1}$$

The overall inter-annotator agreement as the average of $\mathrm{IAA}(x,y)$:

$$\mathrm{IAA} = \frac{\sum_x \sum_{y \neq x} \mathrm{IAA}(x,y)}{2 \cdot \text{number of all couples of different annotators}} \tag{2}$$

From the results we estimate the overall inter-annotator agreement as 66% taking uncertainty into account and 74.2% without it (i.e. accepting e.g. 'y' and 'Y' as the same answer).

[2] http://www.statmt.org/moses

4.3. Success Rates

This section provides the overall results of the four examined MT systems. It also shows, how the success rate depends on and varies with topic domains and text lengths.

First, let us discuss the possibilities of what should be considered a correct answer. The main question is, whether to accept answers 'Y' and 'N' as correct, when the golden answers are 'y' and 'n', or in other words: do we accept an unsure but otherwise correct answer? We decided to accept these answers as correct, as they meant that the reader of the translated text indeed got the information, only not so explicit as it was in the original text.

Another question is how to handle answers 'x' ("can't tell from the text") and 'X' ("don't understand the question"). We took 'x' as an ordinary answer, counting as correct only when the golden answer was also 'x'. Answers 'X' were not taken into account, because they indicated a problem of understanding the question, not the text.

We evaluated the dataset using all the interpretation possibilities and observed differences only in the absolute values but never in overall trends (e.g. the winning MT system). Therefore we present only the judgment strategy described above.

The dataset for evaluation of the four examined MT systems consists of 5588 answers to questions about 1905 text instances as provided by the total of 18 different annotators. 61 answers were not included in final statistics because they were 'X'.

The success rates are computed as follows:

$$\text{Success rate} = \frac{\text{Number of correct answers}}{\text{Number of all answers}} \cdot 100\,\% \tag{3}$$

The overall success rate was 79.5%.

Table 3 shows the success rates for individual MT systems with respect to topic domain and number of sentences in translated texts. Each cell in the table (except the "Overall" row) is based on 115.1 answers on average (standard deviation 26.4, minimum 69, maximum 170 answers).

Tables 4 and 5 show the overall success rates of all examined MT systems with respect to text length and then topic domain.

4.4. Discussion

The results document that the overall success rate is slightly higher than our estimate of intra-annotator and inter-annotator agreement. We have thus probably reached the limits of this type of evaluation. The main good news is that overall, our MT systems allowed to answer nearly 80% of questions correctly. In many practical situations, this success rate can be sufficient. For getting or meeting somewhere, the users should be more cautious as the success rate dropped to 76.59%.

Topic	Text length	Google	CU-Bojar	PC Translator	TectoMT
Directions	1	**81.1%**	72.5%	80.8%	78.4%
	2	77.9%	75.9%	76.4%	**79.3%**
	3	83.3%	68.6%	**85.0%**	79.0%
Meetings	1	**80.2%**	68.4%	64.2%	78.5%
	2	**83.3%**	73.8%	73.8%	75.0%
	3	77.0%	79.5%	**84.7%**	79.5%
News	1	**91.1%**	81.1%	87.8%	89.7%
	2	78.2%	**82.9**%	81.8%	76.7%
	3	75.7%	75.4%	69.7%	**81.1%**
Quizes	1	75.2%	69.9%	82.5%	**84.1%**
	2	78.6%	80.5%	84.4%	**89.1%**
	3	81.1%	76.2%	79.7%	**81.3%**
Overall		80.3%	75.9%	80.0%	**81.5%**

Table 3. Success rates for examined MT systems. Best in each row in bold.

Text length	Success rate
1 sentence	79.93%
2 sentences	79.74%
3 sentences	78.64%

Table 4. Overall success rates for different text lengths

As we see from Tables 4 and 5, the success rates drop only slightly with increasing length of translated texts. The rates of different topic domains are also very close, with the news topic being the most successful. This could be caused by the annotators already knowing some of the information from local media or by the fact that most of the systems are designed to handle "generic text" and compete in shared translation tasks like WMT which are set in the news domain.

Table 6 compares our ranking of systems to various metrics used in the WMT10 evaluation campaign (Callison-Burch et al., 2010). The figures indicate that various manual evaluations provide rather different results. Users of MT systems should therefore evaluate system candidates specifically for the translation task where the systems will eventually serve.

In terms of allowing to correctly answer questions in our examined four domains, TectoMT seems to be the best. It is therefore somewhat surprising that TectoMT was the worst in terms of "Edit deemed acceptable", i.e. the percentage of post-edits of the output carried out without seeing the source or reference that an independent

Topic	Success rate
Directions	78.44%
Meetings	76.59%
News	81.33%
Quizes	80.87%

Table 5. Overall success rates for different topic domains

Metric	Google	CU-Bojar	PC Translator	TectoMT
$\geq$ others (WMT10 official)	**70.4**	65.6	62.1	60.1
$>$ others	49.1	45.0	**49.4**	44.1
Edits deemed acceptable [%]	**55**	40	43	34
Quiz-based evaluation [%]	80.3	75.9	80.0	**81.5**
BLEU	**0.16**	0.15	0.10	0.12
NIST	**5.46**	5.30	4.44	5.10

Table 6. Manual and automatic scores of the MT systems. Best in bold. We report WMT manual evaluations (comparison with other systems and acceptability of post-editing) and the overall result of our quiz-based evaluation.

annotator then validated as to preserve the original input. The discrepancy can have several reasons, e.g. TectoMT performing better on a wider range of text domains than the news domain of WMT10, or our quiz-based evaluation asking about some "core" information from the sentences whereas the acceptability of edits requires all details to be preserved.

Overall, the most fluent output is produced by Google (with respect to the WMT official score based on the percentage of sentences where the system was manually ranked equal or better than other systems as well as with respect to the acceptability of edits). Google ends up being the second in our quiz-based evaluation. PC Translator was often a winner alone, clearly distinct from others, because it scored best in "$>$ others".

The most surprising is the result of CU-Bojar: while second in the official "$\geq$ others", it scores much worse in all other comparisons. CU-Bojar is probably often incomparably similar to the best system but if observed alone, it does not preserve valuable information as good as other systems.

5. Conclusion

In this paper we described a novel technique for manual evaluation of machine translation quality. The presented method, called quiz-based evaluation, is based on

annotators' reading of machine-translated texts and answering questions on information available in original texts. The presented method was used for evaluating four English-to-Czech MT systems participating in WMT10 (Callison-Burch et al., 2010) on short texts in four different topic domains.

The results indicate a completely different order of the evaluated systems compared to both automatic and manual evaluation methods as used in WMT10. The results also suggest that the success rate of machine translation mildly decreases with increasing text length, although our texts were too short (one to three sentences) for a reliable observation. The success rates of various topic domains were also very close, with translations of news being the most successful.

The overall success rate was 79.5%, meaning that on average, machine translation allowed our annotators to answer four of five questions correctly. This suggests a fairly high practical usability of modern machine translation systems.

Acknowledgement

The work on this project was supported by the grants P406/10/P259, P406/11/1499, and the project EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic).

We are grateful to all our student collaborators who provided us with the texts, questions as well as the evaluated annotations.

Bibliography

Bojar, Ondřej and Kamil Kos. 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W10-1705`.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F. Zaidan. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 17–53, Morristown, NJ, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8. URL `http://portal.acm.org/citation.cfm?id=1868850.1868853`.

Hajič, Jan, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4, 2006.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Morristown, NJ, USA, 2007. Association

for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1557769.1557821.

Lavie, A. and M.J. Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115, 2009. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77954763029&partnerID=40&md5=38249c2daa847f4657c08f5f051a1b6e.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073083.1073135. URL http://dx.doi.org/10.3115/1073083.1073135.

Sgall, P., F. Hajičová, and J. Panevová. *The Meaning of Sentence and Its Semantic and Pragmatic Aspects*. Academia, Prague, Czechoslovakia, 1986. ISBN 90-277-1838-5.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 167–170, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1. URL http://portal.acm.org/citation.cfm?id=1626394.1626419.

Žabokrtský, Zdeněk, Martin Popel, and David Mareček. Maximum entropy translation model in dependency-based mt framework. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 207–212, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-1731.

Address for correspondence:
Ondřej Bojar
bojar@ufal.mff.cuni.cz
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25
11800 Praha, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 87–106

Word-Order Issues in English-to-Urdu Statistical Machine Translation

Bushra Jawaid, Daniel Zeman

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

We investigate phrase-based statistical machine translation between English and Urdu, two Indo-European languages that differ significantly in their word-order preferences. Reordering of words and phrases is thus a necessary part of the translation process. While local reordering is modeled nicely by phrase-based systems, long-distance reordering is known to be a hard problem. We perform experiments using the Moses SMT system and discuss reordering models available in Moses. We then present our novel, Urdu-aware, yet generalizable approach based on reordering phrases in syntactic parse tree of the source English sentence. Our technique significantly improves quality of English-Urdu translation with Moses, both in terms of BLEU score and of subjective human judgments.

1. Introduction

Statistical machine translation between languages with significant word order differences and highly inflected morphology on one or both sides is not always straightforward. Linguistic difference between source and target languages makes translation a complex task. English and Urdu, although both belonging to the Indo-European language family, possess quite different characteristics in word order and morphology.

English is read and written from left to right whereas Urdu is read and written from right to left. Both languages differ in morphological and syntactic features. English has a relatively simple inflectional system: only nouns, verbs and sometimes adjectives can be inflected, and the number of possible inflectional affixes is quite small (Jurafsky and Martin, 2000). Urdu on the other hand is highly inflectional and rich in

Corresponding author: zeman@ufal.mff.cuni.cz
Cite as: Bushra Jawaid, Daniel Zeman. Word-Order Issues in English-to-Urdu Statistical Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 87–106.
doi: 10.2478/v10108-011-0007-0.

morphology. In Urdu verbs are inflected according to gender, number and person of the head noun; noun phrases are marked for gender, number and case; and adjectives inflect according to the gender and number of the head noun.

English is a fixed word order language and follows the SVO (Subject-Verb-Object) structure; Urdu is a free word-order language and allows many possible word orderings but the most common sentence structure used by the native speakers is SOV. Also, instead of English prepositions, Urdu nouns and verbs are followed by postpositions.

Example 1 demonstrates the differing word orders on an English-Urdu sentence pair.

(1) English: They understand English and Urdu.
Urdu: وہ انگریزی اوراردو سمجھتے ہیں ۔
Translit.: *wah angrezī aor urdū samjhte heñ .*
Gloss: they English and Urdu understanding are .

A plain phrase-based statistical translation system may not be able to correctly cope with all the differences in grammars of the two languages. The goal of this study is to improve translation quality for the given language pair by making both languages structurally similar before passing the training and test corpora to the SMT system.

(Zeman, 2010) gives an overview of related work for many language pairs. (Bojar et al., 2008) and (Ramanathan et al., 2008) used a rule-based preprocessing approach on English-to-Hindi translation, which is structurally similar to the English-to-Urdu language pair. They achieved significant BLEU score improvement by reordering English sentences in the training and test corpora to make the word order similar to Hindi. In this paper we use a similar scheme based on an effective rule-based transformation framework. This framework is responsible for reordering the source sentence and making its word order as similar to the target language as possible. Our transformation scheme is general and applicable to other language pairs.

2. Overview of the Statistical Machine Translation System

Statistical machine translation (SMT) system is one of the applications of the Noisy Channel Model introduced by (Shannon, 1948) in the information theory. The setup of the noisy channel model of a statistical machine translation system for translating from Language F to Language E works like this: The channel receives the input sentence e of Language E, transforms it ("adds noise") into the sentence f of Language F and sends the sentence f to a decoder. The decoder then determines the sentence $\hat{e}$ of language E that f is most likely to have arisen from and which is not necessarily identical to e.

Thus, for translating from language F to language E the SMT system requires three major components. A component for computing probabilities to generate sentence e,

another component for computing translation probabilities of sentence f given e, and finally, a component for searching among possible foreign sentences f for the one that gives the maximum value for $P(f|e)P(e)$.

Let's treat each sentence as a sequence of words. Assume that a sentence f of language F, represented as $f_1^J = f_1, ..., f_j, ..., f_J$ is translated into a sentence e of language E, and represented as $e_1^I = e_1, ..., e_i, ..., e_I$.

Then, the probability $P(e_1^I|f_1^J)$ assigned to a pair of sentences (f_1^J, e_1^I), is interpreted as the probability that a decoder will produce the output sentence e_1^I given the source sentence f_1^J.

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}}\ P(e_1^I|f_1^J) \tag{2}$$

Equation 2 is also known as Bayes Decision Rule. For translating sentence f_1^J into sentence e_1^I, we need to compute $P(e_1^I|f_1^J)$. For any given probability $P(y|x)$, it can be further broken down using Bayes' theorem.

$$P(e_1^I|f_1^J) = \frac{P(f_1^J|e_1^I)\ .\ P(e_1^I)}{P(f_1^J)} \tag{3}$$

Since we are maximizing over all possible translation hypotheses for the given source sentence f_1^J, Equation 3 will be calculated for each sentence in Language E. But $P(f_1^J)$ doesn't change for each translation hypothesis. So we can omit the denominator $P(f_1^J)$ from the Equation 3.

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}}\ P(f_1^J|e_1^I)\ .\ P(e_1^I) \tag{4}$$

The model of the probability distribution for the first term in Equation 4 ($P(f_1^J|e_1^I)$, likelihood of translation (f, e)) is called Translation Model, and the distribution of $P(e_1^I)$ is called Language Model.

3. The Translation System

The statistical phrase-based machine translation system, Moses[1] (Koehn et al., 2007), is used in this work to produce English-to-Urdu translation. According to (Koehn et al., 2007) "The toolkit is a complete out-of-the-box translation system for academic research. It consists of all the components needed to preprocess data, train the language models and the translation models. It also contains tools for tuning these models using minimum error rate training (MERT) (Och, 2003)".

[1] http://www.statmt.org/moses/

Moses automatically trains the translation models on the parallel corpora of the given language pair. It uses an efficient algorithm to find the maximum probability translation among the exponential number of candidate choices. For this study we have chosen to build the phrase translation table on word 7-grams, unless stated otherwise.

Training is performed using `train-factored-phrase-model.perl` script included in Moses package. Word alignments are extracted using GIZA++[2] (Och and Ney, 2003) toolkit which is a freely available implementation of IBM models for extracting word alignments. Alignments are obtained by running the toolkit in both translation directions and then symmetrizing the two alignments. We use the *grow-diag-final-and* alignment heuristic (Koehn et al., 2003). It starts with the intersection of the two alignments and then adds additional alignment points that lie in the union of the two alignments. This method only adds alignment points between two unaligned words.

For language modeling we use the SRILM toolkit[3] (Stolcke, 2002) with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998). More precisely, we use the SRILM tool `ngram-count` to train our language models.

We use the standard implementation of minimum error rate (MERT) training packed in script `mert-moses.pl`.

4. Data and Their Preprocessing

This section provides a brief overview of the data used in this study. We also summarize some statistics over our corpora. We normalized all Urdu texts to make them usable for training of the translation system. We collected four different parallel corpora of at least three different domains from various sources. In addition, we collected a large monolingual corpus from the Web.

4.1. Parallel Data

We collected the following four English-Urdu parallel corpora to perform our experiments:

- EMILLE (Baker et al., 2002) is a 63 million word corpus of Indic languages which is distributed by the European Language Resources Association (ELRA). The detail of Emille corpus is available from their online manual[4].
- Wall Street Journal (WSJ) texts from the Penn Treebank (Marcus et al., 1999). The English treebank part has been released by the Linguistic Data Consortium (LDC). The parallel Urdu translation is distributed by the Centre for Research in

[2] `http://fjoch.com/GIZA++.html`

[3] `http://www-speech.sri.com/projects/srilm/`

[4] `http://www.lancs.ac.uk/fass/projects/corpus/emille/MANUAL.htm`

Urdu Language Processing (CRULP) under the Creative Commons License. The corpus is freely available online[5] for research purposes. The Urdu translation is a plain text and it is not available in treebank format. Also the whole Treebank-3's translation to Urdu is not yet available, only a subpart of the WSJ section is used in this work.[6]

- Quran translations available on-line.[7]
- Bible translations available on-line. While several English translations of the Bible exist, we were only able to get the parallel translation of the New Testament.[8]

Corpus	Source	SentPairs	Tokens	Vocabulary	Sentence Length	
					μ	σ
Emille	ELRA	8,736	153,519	9,087	17.57	9.87
Penn	LDC	6,215	161,294	13,826	25.95	12.46
Quran	Web	6,414	252,603	8,135	39.38	28.59
Bible	Web	7,957	210,597	5,969	26.47	9.77

Table 1: English parallel corpus size information

Corpus	Source	SentPairs	Tokens	Vocabulary		Sentence Length	
				Raw	Norm	μ	σ
Emille	ELRA	8,736	200,179	10,042	9,626	22.91	13.07
Penn	LDC	6,215	185,690	12,883	12,457	29.88	14.44
Quran	Web	6,414	269,991	8,027	7,183	42.09	30.33
Bible	Web	7,957	203,927	8,995	6,980	25.62	9.36

Table 2: Urdu parallel corpus size information

[5] http://crulp.org/software/ling_resources/UrduNepaliEnglishParallelCorpus.htm

[6] The list of the Penn Treebank files whose parallel Urdu translation is available on-line can be found at http://crulp.org/Downloads/ling_resources/parallelcorpus/Read_me_Urdu.txt and also at http://crulp.org/Downloads/ling_resources/parallelcorpus/read_me_Extended_Urdu.txt. Only the files whose names are listed at these websites are used in this study.

[7] The Quran-English UTF-8 data is downloaded from http://www.irfan-ul-quran.com/quran/english/contents/sura/cols/0/ar/0/ur/0/ra/0/en/1/ and Quran-Urdu UTF-8 data is downloaded from http://www.irfan-ul-quran.com/quran/english/contents/sura/cols/0/ar/0/ur/1/ra/0/en/.

[8] The free King James Bible edition is distributed by "Project Gutenberg Etext". The Bible-English UTF-8 data is downloaded from http://www.gutenberg.org/dirs/etext90/kjv10.txt and the Bible-Urdu UTF-8 data is downloaded from http://www.terakalam.com/

The statistics over the bilingual corpora are summarized in Table 1 and Table 2. The interesting fact in comparison of the two languages is that in all corpora except the Bible the number of Urdu tokens is higher than the number of English tokens. The reason for the different result for the Bible could be different sources of the English and the Urdu part and the linguistic expressiveness adopted by each of the sources. This raises some doubt about the translation quality in the case of the Bible.

Table 2 also summarizes the change in vocabulary size after applying the normalization process (Normalization is discussed in detail in section 4.4). Emille and Penn have smaller loss in vocabulary size after applying normalization, while the Bible corpus loses around 2000 unique words. We can attribute the loss mostly to the wrong usage of diacritic marking that results in multiple (mis-)spellings of the same word. Example 5 shows the varying diacritics on the same word in the unnormalized Bible.

(5) (a) "Who" translated as کون without diacritic marking in bold (correct).
English sentence: And **who** is he that will harm you, if ye be followers of that which is good?
Urdu sentence: اگر تُم نیکی کرنے میں سرگرم ہو تو تُم سے بدی کرنے والا کون ہے؟
Transliteration: *agar tum nekī karne meñ sargaram ho to tum se badī karne wālā kon he?*

(b) "Who" translated as کَون with zabar (ـَ) diacritic mark (correct).
English sentence: And **who** shall be able to stand?
Urdu sentence: اَب کَون ٹھہر سکتا ہے؟
Transliteration: *ab kon ṭhahar saktā he?*

(c) "Who" translated as کُون with *pesh* (ـُ) diacritic mark (incorrect).
English sentence: Then said they unto him, **who** art thou?
Urdu sentence: اُنہوں نے اُس سے کہا تُو کُون ہے؟
Transliteration: *unhoñ ne us se kahā tū kūn he?*

In Example 5 there are three different Urdu forms of the word "who" but only the first two are correct. Example 5 (b) shows the correctly diacriticized form of the word. Since most Urdu literature is written and understandable without diacritics, the form in Example 5 (a) is also correct whereas the form in Example 5 (c) is ill-formed.

The average sentence length varies across the corpora. It ranges from 8 to 39 words on average for English and from 23 to 42 words on average for Urdu. The highest average length is found in Quran while the Emille corpus has the shortest sentences.

In Figure 1 the overall average length of English sentences is about 25 words. It also shows that the Quran corpus contains a few extraordinarily long sentences, with sizes over 240 words. The corresponding graph for Urdu is presented in Figure 2. The overall Urdu average is about 30 words per sentence and again the Quran corpus reaches the extremes of over 260 words.

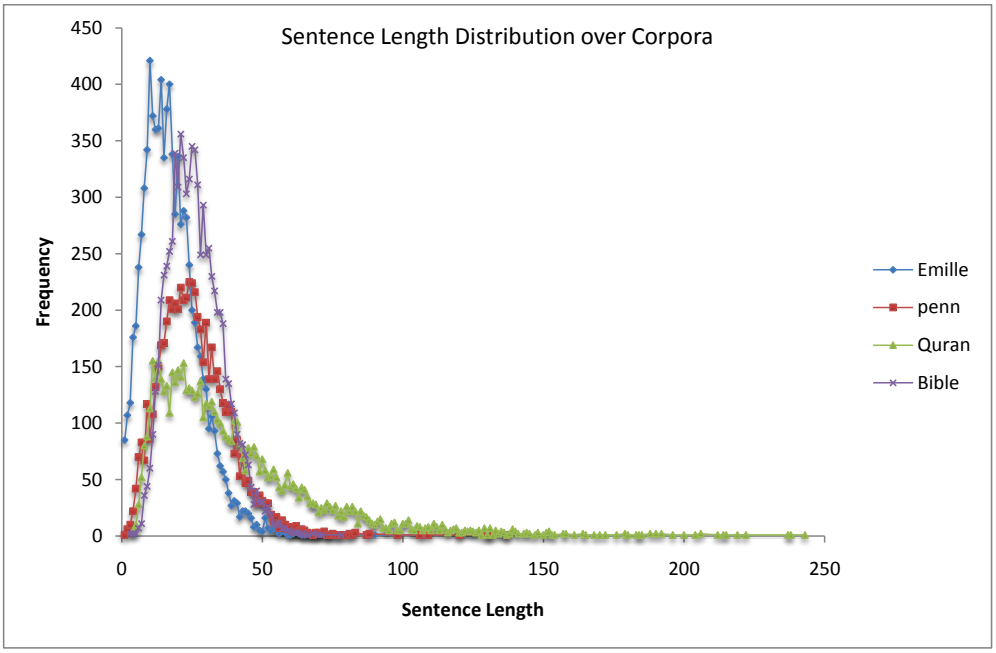


Figure 1: Sentence length distribution over the English side of bilingual corpora

4.2. Monolingual Data

Large monolingual Urdu plain-text corpus has been collected to build the language model that is used by the decoder to figure out which translation output is the most fluent among several possible hypotheses. The main categories of the collected data are News, Religion, Blogs, Literature, Science and Education. The following online sources have been used: BBC Urdu[9], Digital Urdu Library[10], ifastnet[11], Minhaj

[9] http://www.bbc.co.uk/urdu/

[10] http://www.urdulibrary.org/index.php?title=صفحہ_اول

[11] http://kitaben.ifastnet.com/

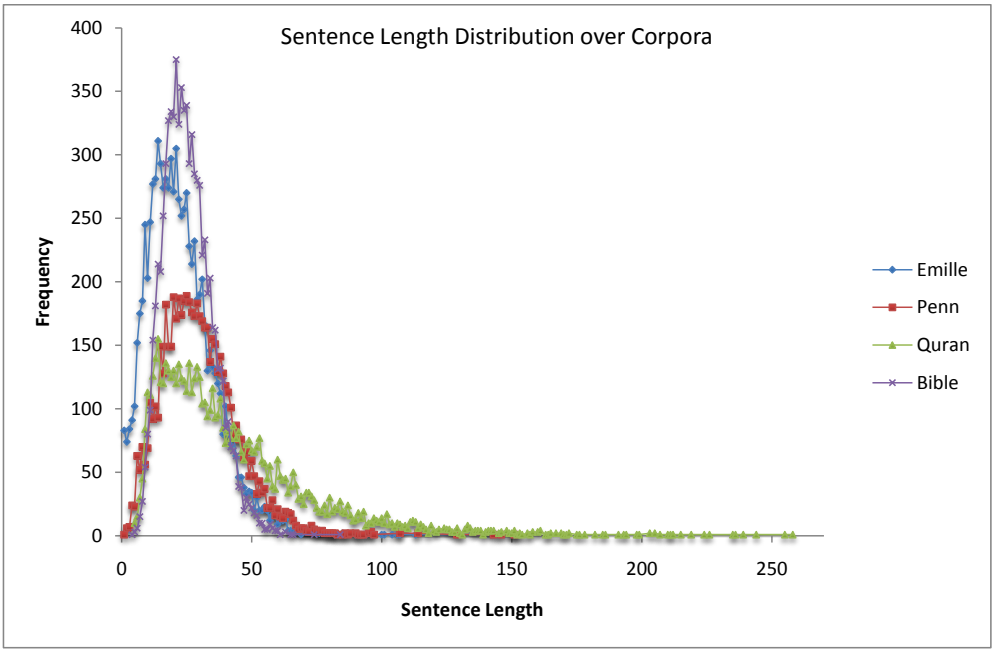


Figure 2: Sentence length distribution over the Urdu side of bilingual corpora

Books[12], Faisaliat[13] and Noman's Diary[14]. The Urdu side of the parallel corpora is also added to the monolingual data.

The monolingual corpus collected for this study contains around 61.6 million tokens distributed in around 2.5 million sentences. These figures cumulatively present the statistics of all the domains whose data is used to build the language model. The language model for this study is trained on a total of 62.4 million tokens in about 2.5 million sentences (after adding the Urdu side of the parallel data).

[12] http://www.minhajbooks.com/urdu/control/Txtformat/یونیکوڈ_کتب.html

[13] http://shahfaisal.wordpress.com/

[14] http://noumaan.sabza.org/

4.3. Data Preparation

Table 3 shows our division of the parallel corpora into training, development and test sets. We use the training data to train the translation probabilities. The development set is used to optimize the model parameters in the MERT phase (the parameters are weights of the phrase translation model, the language model, the word-order distortion model and a "word penalty" to control the number of words on output). The test set, used for final evaluation of translation quality, is left untouched during the training and development phases.

Corpus	Training Size	Development Size	Testing Size	Total Sentence Pairs
Emille	8,000	376	360	8,736
Penn Treebank	5,700	315	200	6,215
Quran	6,000	214	200	6,414
Bible	7,400	300	257	7,957

Table 3: Splitting of parallel corpora in terms of sentence pairs

We divided each corpus by taking the first N_1 sentence pairs for training, then the next N_2 sentences for development and the remaining N_3 sentences for testing. Thus the figures in Table 3 also tell how to reconstruct our data sets from the original corpora.

4.4. Normalization

The data have been edited by a number of different authors and organizations who implement their own writing conventions. For instance, while there is a special set of numerals used with the Arabic/Urdu script, using European "Arabic" digits is also acceptable and published texts differ in what numerals they use. Obviously, a statistical MT system will learn better from a corpus that uses one style consistently. That's why we applied some automatic normalization steps to our corpora. The main inconsistencies are as follows:

- Urdu versus English numerals.
- Urdu versus English punctuation.
- Urdu text with/without diacritics.

An example of an unnormalized sentence from the Penn Treebank and its normalized counterpart is shown in Table 5.

English numerals	0	1	2	3	4	5	6	7	8	9
Urdu numerals	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹

Table 4: Mapping between English and Urdu numerals

Unnormalized Urdu sentence	۱۹۹۷ تک کینسر کا سبب بننے والے ایسبسٹاس کے تقریباً تمام باقیماندہ استعمالات کو غیرقانونی قراردیا جائے گا ۔
Normalized Urdu sentence	1997 تک کینسر کا سبب بننے والے ایسبسٹاس کے تقریبا تمام باقیماندہ استعمالات کو غیرقانونی قراردیا جائے گا ۔
Transliteration	*1997 tak kensar kā sabab banane wāle esbasṭās ke taqrībān tamām bāqīmāndah ist'mālāt ko ǧerqānūnī qarār diyā jāe gā .*
English translation	By 1997, almost all remaining uses of cancer-causing asbestos will be outlawed .

Table 5: Urdu sentence from Penn Treebank before and after normalization

5. Reordering Models

In this section we address selected problems specific to the English-Urdu language pair (though we argue in Section 1 that our conclusions are generalizable at least to related languages, such as other Indo-Aryan languages in place of Urdu). We propose improvement techniques to help the SMT system deal with the problems and introduce tools necessary to apply the techniques.

More specifically, we address the problem of word order differences between the source and the target language. As explained in Section 1, English is SVO language and Urdu follows the SOV word order. In order for an SMT system to be successful, it has to be able to perform long-distance reordering.

A distortion model can be trained with Moses to account for word-order differences. Unfortunately, allowing long-distance reordering makes the search space explode beyond reasonable stack limits (there are too many possible partial hypotheses). The system therefore has to decide prematurely and it is likely to lose good partial hypotheses during the initial stage.

The alternate way we propose is to preprocess the English side (both training and development/test) and try to make its word-order close to the expected word order of the target Urdu text.

5.1. Lexical Reordering in Moses

Moses can learn separate reordering probabilities for each phrase during the training process. The probability is then conditioned on the lexical value of the phrase, and such reordering models are thus also referred to as *lexical.*

Under an *unidirectional* reordering model, Moses learns ordering probability of a phrase in respect to the previous phrase. Three ordering types (M, S, D) are recognized and predicted in an *msd-unidirectional* model:

- *Monotone* (M) means that the ordering of the two target phrases is identical to the ordering of their counterparts in the source language.
- *Swap* (S) means that the ordering of the two phrases is swapped in the target language, i.e. the preceding target phrase translates the following source phrase.
- *Discontinuous* (D) means anything else, i.e. the source counterpart of the preceding target phrase may lie before or after the counterpart of the current phrase but in neither case are the two source phrases adjacent.

Note that the three-state *msd* model can be replaced by a simpler *monotonicity* model in which the S and D states are merged.

A *bidirectional* reordering model adds probabilities of possible mutual positions of source counterparts of the current target phrase and the *following* target phrase (Koehn, 2010).

Finally, a reordering model can be lexically conditioned on just the source phrase *(f)* or both the source and the target phrase *(fe).* By default the *msd-bidirectional-fe* reordering model is used in all our experiments.

5.2. Distance-Based Reordering in Moses

Reordering of the target output phrases is modeled through relative distortion probability distribution $d(start_i, end_{i-1})$, where $start_i$ refers to the starting position of the source phrase that is translated into ith target phrase, and end_{i-1} refers to the end position of the source phrase that is translated into $(i-1)$th target phrase. The reordering distance is computed as $(start_i - end_{i-1})$.

The reordering distance is the number of words skipped (either forward or backward) when taking source words out of sequence. If two phrases are translated in sequence, then $start_i = end_{i-1} + 1$; i.e., the position of the first word of phrase i immediately follows the position of the last word of the previous phrase. In this case, a reordering cost of $d(0)$ is applied (Koehn, 2010). Distance-based model gives linear cost to the reordering distance i.e. movements of phrases over large distances are more expensive.

Whenever we used the distance-based model along with the default bidirectional model, we mention it explicitly.

5.3. Source Parse Tree Transformations

We have used the subcomponent of the rule-based English-to-Urdu machine translation system (RBMT) (Ata et al., 2007) for the preprocessing of the English side of the parallel corpora. The RBMT system belongs to the analysis-transfer-generation class of MT systems. In the analysis step, the source sentence is first morphologically analyzed and parsed. Then, during the transfer step, transformations are applied to the sentence structure found by the parser. The primary goal of the transformation module is to reorder the English sentence according to Urdu phrase ordering rules. The transformation rules are kept separated from the transformation module so that a module can easily be adapted for other target languages. The rules can be easily added and deleted through an XML file. In the generation step we use the open source API of the Stanford Parser[15] to generate the parse tree of the English sentence.

In this work we have modified the transformation module according to our needs. Instead of retrieving the attributes and relationships after the transformation we just linearize the transformed parse tree by outputting the reordered English tokens. Figure 3 shows an English parse tree before and after transformation.

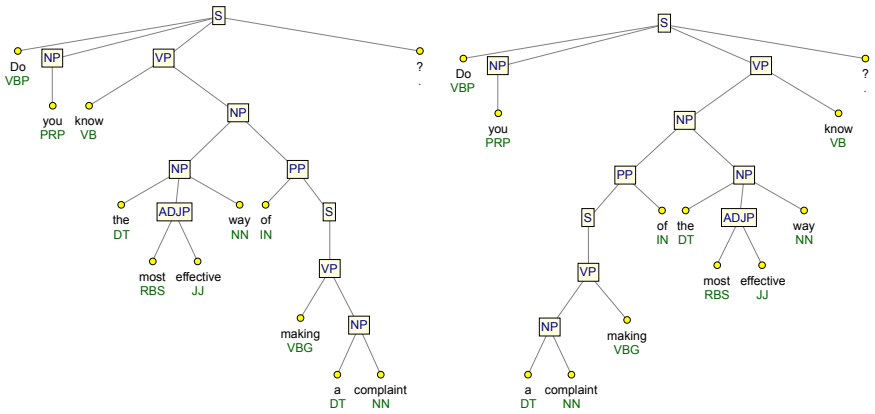


Figure 3: An English parse tree before and after the transformation.

[15] `http://nlp.stanford.edu/software/lex-parser.shtml`
Stanford parser is also available on-line at `http://nlp.stanford.edu:8080/parser/`.

As transformation rules in the RBMT system follow the theoretical model of reverse Panini grammar (Bharati et al., 1995) so, for capturing the most commonly followed word order structures in Urdu we defined a new set of transformation rules. We analyzed the parallel corpora and proposed transformation rules for the most frequent orderings of constituents. A set of nearly 100 transformation rules was compiled. Some instances are shown in Example 6:

(6)
- Prepositions become postpositions.
  Grammar rule: PP → IN NP
  Transformation rule: PP → NP IN
- Verbs come at the end of sentence and ADVP are followed by verbs.
  Grammar rule: S → ADVP VP NP
  Transformation rule: S → NP ADVP VP

The effect of preprocessing the English corpus and its comparison with the distance reordering model are discussed in Section 6.

6. Experiments and Results

Our baseline setup is a plain phrase-based translation model combined with the bidirectional reordering model. Distance-based experiments use both the bidirectional and the distance-based reordering models. (We use the default distortion limit of Moses.) In experiments with preprocessed (transformed) source English data we also use the bidirectional lexical model but not the distance-based model.

All experiments have been performed on normalized target data and mixed[16] language model. In all experiments where normalized target corpus is used, all Urdu data have been normalized, i.e. training data and reference translations of development and test data. See Section 4.4 for a description of the normalization steps.

The translations produced by the different models are illustrated in Table 6. A sentence from the Penn Treebank is presented together with its reference Urdu translation and with translation proposals by three models applying three different approaches to word reordering. Here we would like to mention that the reference translation of the given sentence is not structured well. The reference sentence is split into two comma-separated sections (see the gloss) where a single-clause wording like in the English input would be better. The distance-based system tries to perform the reordering within a window of 6 words whereas our transformation module reached farther and correctly moved the main verb phrase to the end of the sentence.

The other noticeable fact is the correct translation of object phrase "hearings" by our transformation-based system whereas the less sophisticated systems were unable to translate the object noun phrase. The probable reason is that the phrase "The Senate

[16]Mixed language model is the combination of unnormalized monolingual text and normalized target side of the parallel corpora. Although we currently have no explanation, this combination turned out to achieve the best results in terms of BLEU score.

Original sentence	The Senate Banking Committee will begin hearings next week on their proposal to expand existing federal housing programs.
Transformed input	The Senate Banking Committee hearings next week their proposal existing federal housing programs expand to on begin will.
Reference	سینیٹ بنکنگ کمیٹی سماعتیں اگلے ہفتے شروع کرے گی، موجودہ وفاقی ہاؤسنگ پروگراموں کو وسیع کرنے کی ان کی تجویز پر۔
Transliteration	*seneṭ banking kameṭī samāʾteñ agale hafte šurūʾ kare gī , mojūdah wafāqī hāūsing prograāmoñ ko wasīʿ karne kī un kī tajwīz par .*
Gloss	Senate banking committee hearings next week beginning do will, current federal housing programs to wider doing of them of proposal on.
Baseline	سینیٹ بنکنگ کمیٹی شروع کرے گی hearings اگلے ہفتے کے طورپر ان کی تجویز کو وسیع کرنے کے لیے موجودہ وفاقی ہاؤسنگ پروگراموں کے۔
Transliteration	*seneṭ banking kameṭī šurūʾ kare gī hearings agale hafte ke ṭūr par un kī tajwīz ko wasīʿ karne ke līe mojūdah wafāqī hāūsing prograāmoñ ke.*
Distance-based	سینیٹ بنکنگ کمیٹی اگلے ہفتے شروع کرے گی ان کی تجویز پر hearings موجودہ وفاقی ہاؤسنگ پروگراموں کے وسیع کرنے کے لیے ہے۔
Transliteration	*seneṭ banking kameṭī agale hafte šurūʾ kare gī un kī tajwīz par hearings mojūdah wafāqī hāūsing prograāmoñ ke wasīʿ karne ke līe he.*
Transformation-based	سینیٹ کی بنکاری کمیٹی سماعتیں اگلے ہفتے ان کی تجویز پر موجودہ وفاقی ہاؤسنگ پروگراموں کے وسیع کرنے کے لیے پر شروع کرے گی۔
Transliteration	*seneṭ kī bankārī kameṭī samāʾteñ agale hafte un kī tajwīz par mojūdah wafāqī hāūsing prograāmoñ ke wasīʿ karne ke līe par šurūʾ kare gī.*

Table 6: Output translation of baseline, distance-based and transformation-based system.

Banking Committee hearings", also present in training data, had a higher frequency and was learned by the phrase extractor of Moses.

In Urdu, constituents of compound noun phrases in the form "NNP_1 NNP_2" are separated using postpositions as in "NNP_1 IN NNP_2". Due to bringing subject and object phrase closer, much better translation of the subject phrase is retrieved by the transformation-based system, see Example 7. This is a better translation than the mere transliteration used in the reference phrase.

(7)
- *Input:* *Senate* *Banking* *Committee*
  NNP_1 NNP_2 NNP_3
- *Reference:* کمیٹی بنکنگ سینیٹ
  kameṭī *banking* *seneṭ*
  NNP_3 NNP_2 NNP_1
- *Output:* کمیٹی بنکاری کی سینیٹ
  kameṭī *bankārī* *kī* *seneṭ*
  NNP_3 $ADJP_2$ IN NNP_1

According to our analysis the output translation produced by the transformation system is much more accurate then the output produced by the baseline and distance-based models except the additional postposition "پر" *(par)* "on" before the verb phrase "شروع کرے گی" *(šurūʾ kare gī)* "will begin" at the end of the sentence. The reason of placing the postposition before the verb phrase is quite obvious: incorrect placement of the preposition "on" in the transformed input sentence.

In Figure 4 we show the cause of the incorrect placement of the preposition "on" before the verb phrase. In our transformed tree the transformation rule PP → IN NP correctly transformed into PP → NP IN but this transformation actually generated error in the output translation because of the sub-phrase "S" inside the noun phrase (NP). We found out that in all sentences where noun phrases contain "S" or "SBAR" we could automatically remove the sub-phrase node and place it at the end of current transformation rule. For instance in our case the rule PP → NP IN will become PP → NP IN S in transformed tree. The same scheme is also applicable for several other cases where sub-phrases split the constituents of a phrase pair and cause translation errors. The current transformation system doesn't include such sub-phrasal mechanisms yet.

Even the current syntax-aware reordering outperforms both the baseline system and the distance-based reordering model.

In Table 7 we compare the BLEU scores of baseline, distance-based and transformation-based systems. For 3 out of 4 corpora, the transformation-based system is significantly better than both the baseline and the distance-based system. For Quran, the BLEU score decreased from 13.99 (distance-based) to 13.37 (transformation-based).

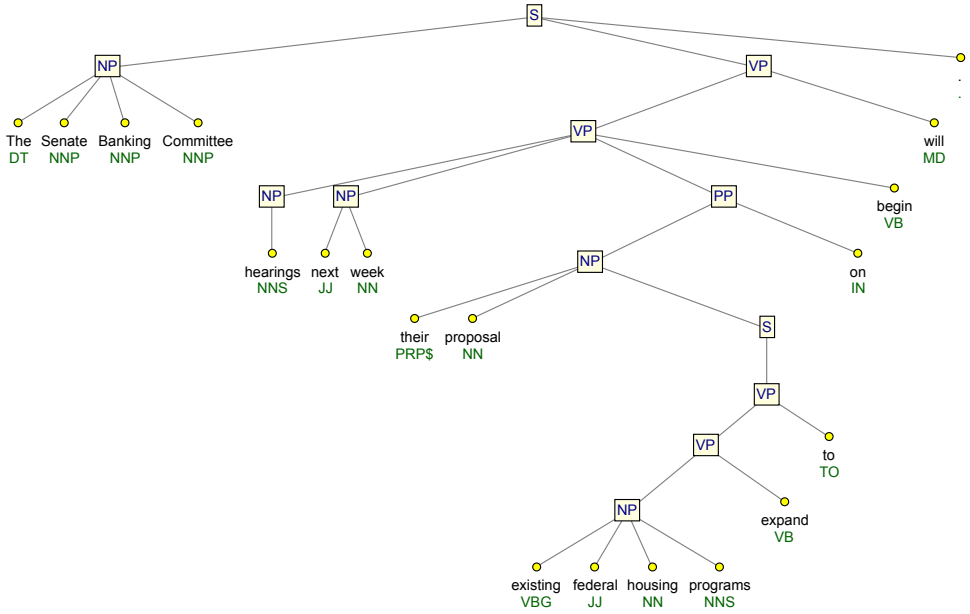


Figure 4: Transformed parse tree of the sentence from Table 6

We suspect that the atypically long sentences of Quran played a role here. Even though the transformations proved to be the best tool available for long-distance reordering, extremely long sentences are more difficult to parse and transformations may have been applied to incorrect parse trees. As an illustration, consider the following English sentence from the Quran:

(8) These people of the book did not dissent among themselves (with regard to believing in the prophethood and messengership of the last messenger [Allah bless him and give him peace] and recognizing his holy status) , until after the clear proof had come to them (of the prophethood of Muhammad [Allah bless him and give him peace] .

There are plenty of parentheses, some of which are not even paired. It is difficult to design transformation rules to handle PRN nonterminals (parentheses) correctly in all situations. We also cannot cover any grammar rule of arbitrarily long right-hand side; instead, heuristics are used to identify subsets of long right-hand sides that could be transformed. Stanford parser analyzes the part *did not dissent among themselves (with regard…), until after…* as

Parallel Data	BLEU Score		
	Baseline	Distance-based	Transformation-based
Emille	21.61	23.59	25.15
Penn Treebank	18.54	22.74	24.07
Quran	13.14	13.99	13.37
Bible	9.39	13.16	13.24

Table 7: Comparison of baseline, distance-based and transformation-based reordering results. All BLEU scores are computed against one reference translation.

VP → VBD NP PP PRN , SBAR
which is heuristically (and incorrectly) transformed to
VP → PRN PP NP VBD , SBAR
The correct transformation for this rule should be
VP → PP NP VBD PRN , SBAR
Also note that the NP label of *not dissent* is a consequence of a tagging error made by the Stanford parser (*dissent* incorrectly tagged as noun). We do not have any easy remedy to these problems; however, see Section 8 for possible directions of future research.

7. Human Evaluation

Automatic evaluation metrics such as the BLEU score are indispensable during system development and training, however, it is a known fact that in some cases and for some language pairs their correlation with human judgment is less than optimal. We thus decided to manually evaluate translation quality on our test data, although due to time and labor constraints we were only able to do this on a limited subset of the data.

We took the Emille test data (360 sentences) and selected randomly a subset of 50 sentences. For each of these sentences, we had five versions: the English source and four Urdu translations: the reference translation and the outputs of the baseline, distance-based and transformation-based systems. We randomized these four Urdu versions so that their origin could not be recognized and presented them to a native speaker of Urdu. Her task was to assign to each Urdu translation one of three categories:

- 2 … acceptable translation, not necessarily completely correct and fluent, but understandable
- 1 … correct parts can be identified but the whole sentence is bad

- 0 … too bad, completely useless, the English meaning cannot be even estimated from it

After restoring the information which sentence came from which model, we counted the sentences in each category. As seen in Table 8, the subjective evaluation confirmed that our transformation approach outperforms automatically learned reordering models.

Category	Reference	Baseline	Distance	Transform
0	1	20	16	12
1	4	20	24	21
2	45	10	10	17

Table 8: Human assessment of translation quality for the reference translation and the outputs of the three systems on a random subset of Emille test data. Category 0 is worst, 2 is best.

8. Conclusion and Future Work

We described our experiments with statistical machine translation from English to Urdu. We collected and normalized significant amounts of parallel and monolingual data from different domains. Then we focused on word order differences and compared two statistical reordering models to our novel syntax-aware, transformation-based preprocessing technique. In terms of automatic evaluation using BLEU score, the transformations outperformed both the lexically conditioned and the distance-based reordering models on all but one corpus. Especially valuable is the fact that we were able to confirm the improvement by subjective human judgments, although we were only able to perform a small-scale evaluation.

We identified the following open problems which could guide the future work:

- Sub-phrasal rules as sketched in the discussion to Figure 4 might improve the transformation results.
- Very long sentences with many parentheses (a specialty of the Quran corpus) are hard to parse, transform and translate. A *divide-et-impera* approach could be explored here: e.g. extracting the parentheses from the source text and translating them separately could address both computational complexity and translation quality at the same time.
- Arbitrarily long rules of the treebank grammar cannot be covered by a predefined set of transformations. In theory, the grammar could be automatically converted and the number of right-hand-side symbols limited in a way similar to standard algorithms of creating a *normal form* of a grammar. However, it is not clear how such a normalization algorithm should be designed. It should

not just mechanically split right-hand sides after the n-th nonterminal because it could separate two symbols that together triggered a transformation.
- Tagging and parsing errors may negatively affect the accuracy of the transformations. Their precise impact should be evaluated and possibly compared to other parsers. Parser combination could improve the results.

Besides word order, Urdu and English also differ in morphology, a fact that has been mostly ignored in the present study. It would also be interesting to see how factored translation models can improve generation of various word forms on the Urdu side.

Acknowledgements

The work on this project was supported by the grants MSM0021620838 of the Czech Ministry of Education, P406/11/1499 of the Czech Science Foundation and the "specific university research" project 261314/2010.

Bibliography

Ata, Naila, Bushra Jawaid, and Amir Kamran. Rule based English to Urdu machine translation. In *Proceedings of Conference on Language and Technology (CLT'07)*. University of Peshawar, 2007.

Baker, Paul, Andrew Hardie, Tony McEnery, Hamish Cunningham, and Rob Gaizauskas. EMILLE, a 67-million word corpus of Indic languages: Data collection, mark-up and harmonisation. In *Proceedings of the 3rd Language Resources and Evaluation Conference, LREC 2002*, pages 819–825. ELRA, 2002. URL `http://gandalf.aksis.uib.no/lrec2002/pdf/319.pdf`.

Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. *Natural Language Processing, a Paninian Perspective*. Prentice Hall of India, New Delhi, India, 1995.

Bojar, Ondřej, Pavel Straňák, and Daniel Zeman. English-Hindi translation in 21 days. In *Proceedings of the 6th International Conference on Natural Language Processing (ICON-2008) NLP Tools Contest*, pages 4–7, 2008.

Chen, Stanley F. and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Technical report TR-10-98, Computer Science Group*, Harvard, MA, USA, August 1998. Harvard University. URL `http://research.microsoft.com/en-us/um/people/joshuago/tr-10-98.pdf`.

Jurafsky, Daniel and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, Upper Saddle River, NJ, 2000. ISBN 0-13-095069-6.

Kneser, Reinhard and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, Los Alamitos, California, USA, 1995. IEEE Computer Society Press.

Koehn, Philipp. *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK, 2010. ISBN 978-0-521-87415-1.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073445.1073462.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Praha, Czechia, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P07-2045`.

Marcus, Mitchell P., Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3, 1999.

Och, Franz Josef. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075117.

Och, Franz Josef and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

Ramanathan, Ananthakrishnan, Pushpak Bhattacharyya, Jayprasad Hegde, Ritesh M. Shah, and M. Sasikumar. Simple syntactic and morphological processing can help English-Hindi statistical machine translation. In *International Joint Conference on NLP (IJCNLP08)*, 2008.

Shannon, Claude E. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

Stolcke, Andreas. SRILM – an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

Zeman, Daniel. Using TectoMT as a preprocessing tool for phrase-based statistical machine translation. In Sojka, Petr, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue. 13th International Conference, TSD 2010, Brno, Czech Republic, September 6-10, 2010. Proceedings*, volume 6231 of *Lecture Notes in Computer Science*, pages 216–223, Berlin / Heidelberg, 2010. Springer. ISBN 978-3-642-15759-2.

Address for correspondence:
Daniel Zeman
`zeman@ufal.mff.cuni.cz`
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25
CZ-11800 Praha, Czechia

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 107–110

NOTES

Frederick Jelinek's Obituary

Jan Hajič

Prof. Frederick Jelinek, dr.h.c., Julian Sinclair Smith Professor at the Whiting School of Engineering at the Johns Hopkins University and the director of JHU's Center for Language Speech and Processing, died unexpectedly at his workplace on Sept. 14, 2010. Prof. Jelinek is survived by his wife Milena Jelinek, professor at Columbia University, son and daughter William and Hannah, three grandchildren and his sister Susan Abramowitz.

Prof. Frederick ("Bedřich" in Czech) Jelinek was born Nov. 18, 1932 in the former Czechoslovakia; his father Vilém was a dentist in a small city of Kladno, near Prague, the capital of Czechoslovakia (now the Czech Republic). The family was half Jewish; his mother was born to Czech parents in Switzerland. Thus, during the Nazi occupation of Czechoslovakia, at the time of "Protektorat Böhmen und Mähren", 1939-1945) they experienced very difficult times, as many Jews did at the time. In 1941, they even had to leave their home city and move to Prague. His father, who had planned emigration in the early days of German rule but – tragically – decided to stay, was eventually deported to Theresin, a Jewish ghetto north of Prague. He died there because of a typhus epidemic in the last days of the World War II.

Bedřich Jelinek then entered a Czech high school. He actually had trouble getting through, due to three missing years of formal education that was stripped from him, as from many others, by various anti-Jew Nazi decrees. After 1948, when the Communists came to power following the well-known "February coup" in Czechoslovakia, his mother sagaciously decided to leave the country. One of the reasons was also that the revolutionary organization of Communist Youth would not allow her son to even take the high school graduation exam. Thanks to her Swiss origins, they were easily allowed into the United States and they settled in New York. Frederick Jelinek then started evening engineering courses at the City College of New York, despite being interested more in becoming a lawyer. However, as he also recalled in his acceptance speech of the honorary doctorate at the Charles University in Prague in 2001, he thought that his foreign accent would make him a less successful lawyer and that also

it took much longer to get the degree (and consequently to earn money for living) than in engineering. Today, we can only imagine how a good lawyer he would have been, if he were equally successful at the bar as he has been in his "forced" engineering career.

After two years at the City College, he has received a stipend from the Committee for Free Europe. As a part of the deal, he had to promise them to help rebuild Czechoslovakia once free again. Frederick Jelinek then started regular classes at MIT, where he met Claude Shannon and embarked on the study on theory of information, happy that the goal of this branch of science is "not to build physical systems". As we know now, it was the beginnings of the information theory being applied to other branches of science. However, it was not yet applied to linguistics, even though we can trace some connections there, too: Frederick Jelinek, after he graduated in 1956 and started his doctorate in the same field, was often talking to Roman Jakobson, a Russian linguist with close ties to Czechoslovakia, who worked at both Harvard and MIT. Jakobson also arranged for a stipend for Frederick Jelinek's wife, Milena, to study at Noam Chomsky's department once she was allowed out of Czechoslovakia in 1961 as a measure of "friendship" of the Czechoslovak government to John F. Kennedy after he was elected U.S. president. After he got his Ph.D. from MIT, Frederick Jelinek joined Cornell University as a professor. He already wanted to start pursuing the connection between linguistics and information theory there, but the professor who was supposed to work on this topic with him there pulled out of the field.

The turning point came in 1972, ten years after he joined Cornell: as part of his unpaid 3 months as a professor, he accepted a position at IBM T. J. Watson Research Center in Yorktown Heights in New York. IBM was then starting to look into the speech recognition problem, and after the sudden departure of the group manager, they offered the position to him. Frederick Jelinek then stayed at this "temporary" position for two years, after which he had to leave Cornell completely but he kept his IBM position. He was the head of the Speech group for the next 19 years, the years that changed the field of computational linguistics the most in its entire history.

The IBM speech group, first located in Yorktown and then in Hawthorne, New York, consisted of almost no linguists: rather, the researchers had been educated either in engineering, information theory, or in physics. They were thus skeptical to the linguistic experts who were devising speech recognition systems at that time. As Frederick Jelinek recalls, the key to their success was probably their "naïve approach to this problem". They threw all the then-current methods out and started from scratch, applying information theory, statistical methods and machine learning to the speech recognition problem and later to machine translation. After almost twenty years since then, we now know the results of this "naïve" approach – they have not been surpassed yet. Moreover, all commercial large vocabulary speech recognizers now on the market use these methods with only relatively minor modifications.

In 1993 Frederick Jelinek joined Johns Hopkins University in Baltimore, Maryland, and became the director of the Center for Language and Speech processing at the

Whiting School of Engineering. While at Johns Hopkins University, he was awarded many NSF, DARPA and other grants. Among them, there was a series of grants that stands out: the grants for the organization of the now famous (and often emulated) JHU Summer Workshop (officially, the "Workshop on Language Engineering for Students and Professionals Integrating Research and Education"). It is an 8-week (including two weeks of a Summer School for undergraduate students selected world-wide) labor-intensive event, where carefully peer-selected projects are being worked on by two to four teams of professors, researchers, graduate and undergraduate students. It is hard to find a well-known researcher in the field of speech recognition or computational linguistics who has not been there at least once during her or his career.

After 1989, the year of the fall of the Berlin Wall and the political changes in Czechoslovakia (also known as the "Velvet Revolution"), he started paying off his promise to his MIT stipend Committee: he started to visit Czechoslovakia (then Czech Republic) often, and invited first Czechs to his IBM team to work on both speech recognition and machine translation. The author was the first one to do so, soon followed by several others, who are now working at IBM or the academia both in the Czech Republic and in the U.S. He also taught in Prague, both at the Charles University and at the Technical University. He arranged for a gift to the Technical University in Prague, and then helped to get his managers to agree to keep part of the Watson speech recognition and development team in Prague, where they reside until today. He also collaborated with Charles University later, inviting professors, postdocs, and students in various capacities to his new place of work after he had joined the Johns Hopkins University in 1993. In 2001, he spent his sabbatical year in the Czech Republic, working and lecturing at the Institute of Formal and Applied Linguistics, which is part of the Computer Science School of Charles University in Prague. At that time, he also received his honorary doctorate from Charles University. He was then coming often to visit conferences, for example the "Text, Speech and Dialog" (TSD) conference organized jointly by the University of West Bohemia in Pilsen and the Masaryk University in Brno, of which he was the honorary chairman of the organizing committee. He continued teaching intensive courses in speech recognition at Charles University and elsewhere, and he was also sending his students to spend some time in Prague under the NSF PIRE project he headed. Recently, he also started intensive collaboration with the Technical University in Brno, also in the Czech Republic.

We in Prague talked to him, regretfully only very briefly, just before his return from the TSD conference back to Baltimore this past September. No one knew at the moment that there are only three more days left for him in this world. No one could imagine that we (or anybody else) will never see him or talk to him again. I am afraid that I cannot fully imagine it even today.

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Prague, Czech Republic

Note: The obituary is reprinted here from the fall 2010 EACL Newsletter with the kind permission of the Executive Board of the European Chapter of the ACL.

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011

INSTRUCTIONS FOR AUTHORS

Manuscripts are welcome provided that they have not yet been published elsewhere and that they bring some interesting and new insights contributing to the broad field of computational linguistics in any of its aspects, or of linguistic theory. The submitted articles may be:

- long articles with completed, wide-impact research results both theoretical and practical, and/or new formalisms for linguistic analysis and their implementation and application on linguistic data sets, or
- short or long articles that are abstracts or extracts of Master's and PhD thesis, with the most interesting and/or promising results described. Also
- short or long articles looking forward that base their views on proper and deep analysis of the current situation in various subjects within the field are invited, as well as
- short articles about current advanced research of both theoretical and applied nature, with very specific (and perhaps narrow, but well-defined) target goal in all areas of language and speech processing, to give the opportunity to junior researchers to publish as soon as possible;
- short articles that contain contraversing, polemic or otherwise unusual views, supported by some experimental evidence but not necessarily evaluated in the usual sense are also welcome.

The recommended length of long article is 12–30 pages and of short paper is 6-15 pages.

The copyright of papers accepted for publication remains with the author. The editors reserve the right to make editorial revisions but these revisions and changes have to be approved by the author(s). Book reviews and short book notices are also appreciated.

The manuscripts are reviewed by 2 independent reviewers, at least one of them being a member of the international Editorial Board.

Authors receive two copies of the relevant issue of the PBML together with the original pdf files.

The guidelines for the technical shape of the contributions are found on the web site `http://ufal.mff.cuni.cz/pbml.html`. If there are any technical problems, please contact the editorial staff at `pbml@ufal.mff.cuni.cz`.