Yandex School of Data Analysis approach to English-Turkish translation at WMT16 News Translation Task

Anton Dvorkovich[1,2], **Sergey Gubanov**[2], **and Irina Galinskaya**[2]
{dvorkanton,esgv,galinskaya}@yandex-team.ru

[1] Yandex School of Data Analysis, 11/2 Timura Frunze St., Moscow 119021, Russia
[2] Yandex, 16 Leo Tolstoy St., Moscow 119021, Russia

Abstract

We describe the English-Turkish and Turkish-English translation systems submitted by Yandex School of Data Analysis team to WMT16 news translation task. We successfully applied hand-crafted morphological (de-)segmentation of Turkish, syntax-based pre-ordering of English in English-Turkish and post-ordering of English in Turkish-English. We perform desegmentation using SMT and propose a simple yet efficient modification of post-ordering. We also show that Turkish morphology and word order can be handled in a fully-automatic manner with only a small loss of BLEU.

1 Introduction

Yandex School of Data Analysis participated in WMT16 shared task "Machine Translation of News" in Turkish-English language pair.

Machine translation between English and Turkish is a challenging task, due to the strong differences between languages. In particular, Turkish has rich *agglutinative morphology*, and the *word order* differs between languages (SOV in Turkish, SVO in English).

To deal with these dissimilarities, we preprocess both source and target parts of the parallel corpus before training: we perform morphological segmentation of Turkish and reordering of English into Turkish word order, aiming to achieve a monotonous one-to-one correspondence between tokens to aid SMT.

Since we changed the target side of the parallel corpus, at runtime we had to do post-processing: desegmentation of Turkish for EN-TR and post-ordering of English words for TR-EN. We employ additional SMT decoders to solve both tasks, which results in two-stage translation.

For morphological segmentation and English-to-Turkish reordering we tried both rule-based/supervised and fully unsupervised approaches.

2 Data & common system components

In our two systems (Turkish-English and English-Turkish) we used several common components described below.

The specific application of these tools varies for Turkish-English and English-Turkish systems, so we discuss it separately in Sections 4 and 3.

2.1 Phrase-based translator

We used an in-house implementation of phrase-based MT (Koehn et al., 2003) with Berkeley Aligner (Liang et al., 2006) and MERT tuning (Och, 2003).

2.2 English syntactic parser

We used an in-house transition-based English dependency parser similar to (Zhang and Nivre, 2011).

2.3 English-to-Turkish reorderers

We used two different reorderers that put English words in Turkish order. Both reorderers need an English dependency parse tree as input.

Rule-based reorderer modifies parse trees using rules similar to Tregex (Levy and Andrew, 2006), adapted to dependency trees[1]. We used a set of about 70 hand-crafted rules, an example of a rule is given in Figure 1.

```
w1 role 'PMOD'
   and .--> (w2 not role 'CONJ')
::
move group w1 before node w2;
```

Figure 1: Sample dependency tree reordering rule

[1] Our dependency tree reordering tool is available here: https://github.com/yandex/dep_tregex

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 281–288,
Berlin, Germany, August 11-12, 2016.

Automatic reorderer uses word alignments on a parallel corpus to construct reference reorderings, and then trains a feedforward neural-network classifier which makes node-swapping decisions (de Gispert et al., 2015).

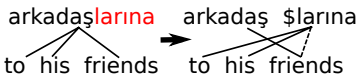
2.4 Turkish morphological analyzers

We used an in-house finite state transducer similar to (Oflazer, 1994) for Turkish morphological tagging, and structured perceptron similar to (Sak et al., 2007) for morphological disambiguation.

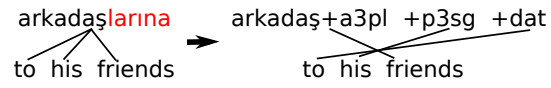
As an alternative, we trained our implementation of unsupervised morphology model, following (Soricut and Och, 2015), with a single distinctive feature: in each connected component C of the morphological graph, we select the lemma as $\operatorname{argmax}_C (\log f(w) - \alpha \cdot l(w))$, where $l(w)$ is word length and $f(w)$ is word frequency[2]. This is a heuristic, justified by the facts, that (1) lemma tends to be shorter than other surface forms of a word, and (2) $\log f(w)$ is proportional to $l(w)$ (Strauss et al., 2007). We also make use of morphology induction for unseen words, as described in the original paper. The automatic method requires no disambiguation and yields no part-of-speech tags or morphological features.

2.5 Turkish morphological segmenter

We used three strategies for segmenting Turkish words into less-sparse units. The "simple" strategy splits a word into lemma and chain of affixes. The latter is chosen as suffix of the surface form, starting from $(l+1)$-th letter, where l is lemma's length.

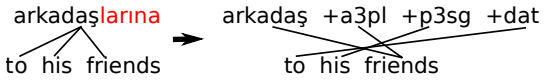


The "rule-based" strategy uses hand-crafted rules similar to (Oflazer and El-Kahlout, 2007), (Yeniterzi and Oflazer, 2010) or (Bisazza and Federico, 2009) to split word into lemma and groups of morphological features, some of which might be attached to lemma. Rules are designed to achieve a better correspondence between Turkish and English words. This strategy requires morphological analyzer to output features as well as lemma.



[2]We used $\alpha = 0.6$ throughout our experiments.

The "aggressive rule-based" strategy, in addition, forcefully splits all features attached to the lemma into a separate group.



2.6 NMT reranker

Finally, we used a sequence-to-sequence neural network with attention (Bahdanau et al., 2014) as a feature for 100-best reranking. We used hidden layer and embedding sizes of 100, and vocabulary sizes of 40000 (the Turkish side was morphologically segmented).

2.7 Data

For training translation model, language models, and NMT reranker, we used only the provided constrained data (SETIMES 2 parallel Turkish-English corpus, and monolingual Turkish and English Common Crawl corpora).

Throughout our experiments, we used the BLEU (Papineni et al., 2002) on provided devset (news-dev2016) to estimate the performance of our systems, tuning MERT on a random sample of 1000 sentences from the SETIMES corpus (these sentences, to which we refer as "the SETIMES subsample", were excluded from training data). For the final submissions, we tuned MERT directly on news-dev2016.

Due to our setup, we provide BLEU scores on news-dev2016 for our intermediate experiments and on news-test2016 for our final systems.

3 Turkish-English system

3.1 Baseline

For a baseline, we trained a standard phrase-based system: Berkeley Aligner (IBM Model 1 and HMM, both for 5 iterations); phrase table with up to 5 tokens per phrase, 40-best translation options per source phrase, and Good-Turing smoothing; 5-gram lowercased LM with stupid backoff and pruning of singleton n-grams due to memory constraints; MERT on the SETIMES subsample; simple reordering model, penalized only by movement distance, with distortion limit set to 16.

We lowercased both the training and development corpora, taking into account Turkish specifics: $I \to ı$, $İ \to i$.

Baseline system achieves 10.84 uncased BLEU on news-dev2016 (here and on, we ignore case in BLEU computation).

#	System description	BLEU (uncased), dev[3]	BLEU (uncased), test[3]
1	Baseline, phrase-based	11.68	11.50
2	(1) + automatic morph., simple seg.	12.16	-
3	(1) + FST/perceptron morph., simple seg.	11.75	-
4	(1) + FST/perceptron morph., rule-based seg.	12.93	-
5	(1) + FST/perceptron morph., aggressive rule-based seg.	14.06	-
6	(5) + "reordered" post-ordering, rule-based reorderer	14.24	-
7	(5) + "translated" post-ordering, rule-based reorderer	15.13	-
8	(2) + "translated" post-ordering, automatic reorderer	13.43	13.39
9	(7) + NMT reranking in first stage	15.49	**15.12**

Table 1: Our TR-EN setups on news-dev2016 and news-test2016 (submitted system in bold)

3.2 Morphological segmentation

In Turkish-to-English translator we directly applied Turkish morphological segmenters (see Section 2.5) as an initial step in the pipeline (Oflazer and El-Kahlout, 2007; Bisazza and Federico, 2009).

The effect of different morphological tagging and segmentation methods is shown in Table 1.

FST/perceptron analyzer with aggressive rule-based segmentation (run #5) turned out to be the most successful method, bringing +2.60 BLEU.

Our segmenters split Turkish words into lemmas and auxiliary tokens like `$ini` or `+a3sg`. To account for the increased number of tokens on Turkish side, we increased the length of a target phrase from 5 to 10 (but still allowing only up to 5 non-auxiliary tokens in a phrase). In order to further decrease sparsity we also removed all diacritics from the intermediate segmented Turkish. Possible ambiguity in translations, caused by this, is handled by English LM.

For a rule-based segmentation we note that it is beneficial to aggressively separate away lemma and morphological features that would normally be attached to it (that is, if we acted according to the rules). We think the reason for this is the presence of errors and non-optimal decisions in our segmentation rules, but we still consider the extra split helpful:

- If we do the extra split, a wordform is segmented into a lemma and several auxiliary tokens, so if we have seen just the lemma, we might still translate the unseen wordform correctly.
- An excessive segmentation does not really hurt a phrase-based system, as shown by (Chang et al., 2008).

3.3 Post-ordering

It is not possible to directly apply English-to-Turkish reorderer as a preprocessing step in this translation direction, and we also counld not construct a Turkish-to-English reorderer (due to the absence of Turkish parser).

Instead, we reordered the target side of the parallel corpus on the training phase using the rule-based reorderer described in Section 2.3, and employed a second-stage translator to restore English word order at runtime, following (Sudoh et al., 2011).

As shown in Figure 2, the first, "monotonous translation" stage is trained to translate from Turkish to English that was reordered to the Turkish order[4], and the second, "reordering" stage is trained to translate from reordered English to normal English, relying on the LM and baseline reordering inside the phrase-based decoder.

[3]We tune on the SETIMES subsample for "dev" column, and on news-dev2016 for "test" column. So the same line lists the results for two sets of MERT coefficients.

[4]This does not mean we completely disable the baseline reordering mechanism in the decoder on this stage; that would have made sense only if (a) our English-to-Turkish reorderer was perfect and (b) if the two languages could be perfectly aligned using just word reordering. Obviously, neither of those is the case.

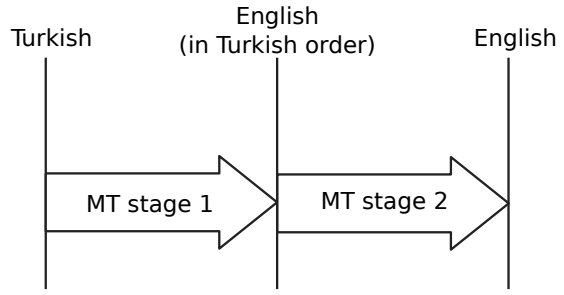


Figure 2: Two-stage post-ordering

Figures 3 and 4 illustrate the training of two-stage postordering systems. We explore two options for the training of the second, "reordering" stage: as the source-side, we can either use (a) the reordered English sentences, or (b) Turkish sentences translated to reordered English with first-stage translator.

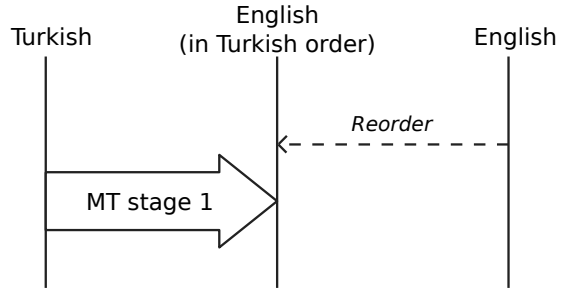


Figure 3: Training the "monotonous translation" stage of post-ordering system

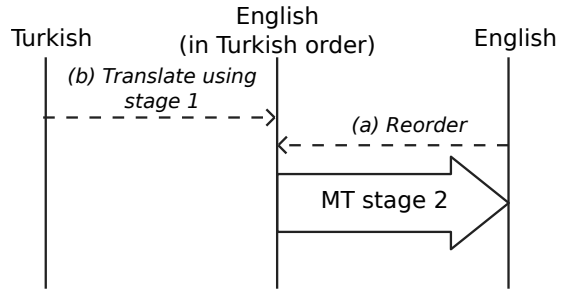


Figure 4: Two options for training the "reordering" stage of post-ordering system

The two decoders have two sets of MERT coefficients. We tune them jointly and iteratively: first, we tune the first-stage decoder (with second-stage coefficients fixed), optimizing BLEU of the whole-system output, then we tune the second-stage decoder (with first-stage coefficients fixed), again optimizing the whole-system BLEU, and so on.

As shown in Table 1, the best results are achieved using "translated Turkish" for training the second-stage translator, yielding an additional +1.60 BLEU.

3.4 NMT reranking

Finally, we enhanced the first-stage translator with a 100-best reranking which uses decoder features and a neural sequence-to-sequence network described in Section 2.6. To train the network, we used the same corpus used to train the first-stage PBMT translator (incorporating Turkish segmentation and English reordering).

NMT reranking yields an additional +0.47 BLEU score.

3.5 Final system

The complete pipeline of our submitted system is shown in Figure 5.

We selected the setup that performed best during experiments (#9 in Table 1), and re-tuned it on the development set; for contrastive runs we also re-tuned baseline and "fully automatic" systems (#1 and #8 respectively). See Table 1 for results.

Our best setup reaches 15.17 BLEU, which is a +3.17 BLEU improvement over the baseline.

The system without the hand-crafted rules achieves a lower improvement of +1.89 BLEU, which is a nice gain nevertheless. Comparing runs #2 and #3, we see that the decrease in BLEU is not due to the quality of morphological analysis; comparing runs #3 and #5, we see that the difference in quality is purely due to the segmentation scheme.

4 English-Turkish system

4.1 Baseline

As a baseline, we trained the same phrase-based system as in Section 3.1 (except we did not prune singleton n-grams in the Turkish language model).

Baseline system achieves 8.51 uncased BLEU on news-dev2016.

4.2 Pre-ordering

We directly apply English-to-Turkish reorderers described in Section 2.3 as a pre-processing step in the phrase-based MT pipeline, like e.g. (Xia and McCord, 2004; Collins et al., 2005). Results are shown in Table 2

The rule-based reorderer earns +1.65 BLEU against the baseline (run #2), so we selected it as a

#	System description	BLEU (uncased), dev[3]	BLEU (uncased), test[3]
1	Baseline, phrase-based	8.51	9.26
2	(1) + rule-based preordering	10.16	-
3	(1) + automatic preordering	9.79	-
4	(2) + deseg. from FST/perceptron morph. & rule-based seg.	11.32	**11.10**
5	(3) + deseg. from automatic morph. & simple seg.	10.41	11.03

Table 2: Our EN-TR setups on news-dev2016 and news-test2016 (submitted system in bold)

base for further improvements. The automatic reorderer performs almost as well as the rule-based (-0.37 BLEU).

4.3 Desegmentation

We decided to battle data sparsity on target side using morphological desegmentation: translate from English to segmented Turkish, then desegment the output.

After experiments in Section 3.2 we decided to use an aggressive rule-based segmenter. First-stage translator makes mistakes, sometimes producing wrong morphemes and/or morphemes in an incorrect order. To manage that, we decided to make desegmentation using machine translation (conceptually similar to post-ordering).

For training MT desegmenter we need only a monolingual corpus, so we can use more data than we used for training the first-stage translator. We concatenated the Turkish part of SETIMES parallel corpus with a random sample of 2 million sentences from Common Crawl monolingual Turkish corpus for training the MT desegmenter.

Like for segmentation, we increased the phrase length on the segmented Turkish side for both translation stages (see Section 3.2). We also removed diacritics from the segmented Turkish; natural Turkish language model employed on the desegmentation stage works like a context-aware restorer of diacritics. Like for post-ordering, we tune MERT coefficients of our two-stage translator jointly (see Secion 3.3).

Our desegmentation scheme yielded +1.16 BLEU (run #4).

4.4 Final system

The complete pipeline of our submitted system is shown in Figure 6.

For the submission, we re-tuned our best run #4 on news-dev2016; for contrastive runs we also re-tuned baseline and "fully-automatic" systems (#1 and #5 respectively). See Table 2 for results.

Our best setup reaches 11.10 BLEU on the test-set, which is a +1.84 BLEU improvement over the baseline.

An almost equal BLEU improvement of +1.77 can still be achieved even if we do not use hand-crafted rules for reordering or segmentation.

5 Conclusions

We successfully applied data preprocessing for improving MT quality, which resulted in +1.84 BLEU improvement on English-Turkish and +3.17 BLEU on Turkish-English. Handling Turkish morphology via segmentation/desegmentation and handling Turkish SOV word order via pre-ordering/post-ordering both yield improvements of comparable importance.

We were able to avoid the manual construction of a desegmenter. We also proposed an efficient modification of post-ordering: to train the "post-ordering" stage by using the translations of the first stage. We believe that is benefitial due to a better between-stage consistency: what second-stage translator sees during training, it sees at runtime.

We also show that unsupervised methods for segmentation and reordering yield a comparable gain of +1.77 BLEU on English-Turkish and a lower gain +1.89 BLEU on Turkish-English. We believe that the lower gain on Turkish-English is due to the simpler segmentation scheme (not due to the lower quality of unsupervised morphology), but a further analysis is needed to understand why such scheme is sufficient for translating in reverse direction.

Our system turned out to be a quite long segmentation/translation/reordering pipeline. That suggests 3 different directions for the future work:

- Further improve the components of the pipeline.
- Replace "translation" components of the pipeline with another kind of decoder (e.g. NMT).
- Abandon the pipeline and consider joint methods, in order to beat error propagation.

6 Acknowledgements

We thank Valentin Goussev and Mariya Shmatova for the linguistic expertise. We also thank Alexey Baytin for the helpful advice.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Arianna Bisazza and Marcello Federico. 2009. Morphological pre-processing for Turkish to English statistical machine translation. In *IWSLT*, pages 129–135.

Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pages 224–232. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics.

Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2015. Fast and Accurate Preordering for SMT using Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1012–1017.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234. Citeseer.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kemal Oflazer and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32. Association for Computational Linguistics.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and linguistic computing*, 9(2):137–148.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In *Computational Linguistics and Intelligent Text Processing*, pages 107–118. Springer.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proc. NAACL*.

Udo Strauss, Peter Grzybek, and Gabriel Altmann. 2007. Word length and word frequency. In *Contributions to the science of text and language*, pages 277–294. Springer.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Post-ordering in statistical machine translation. In *Proc. MT Summit*.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 508. Association for Computational Linguistics.

Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 454–464. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 188–193. Association for Computational Linguistics.

A Pipelines of the submitted systems

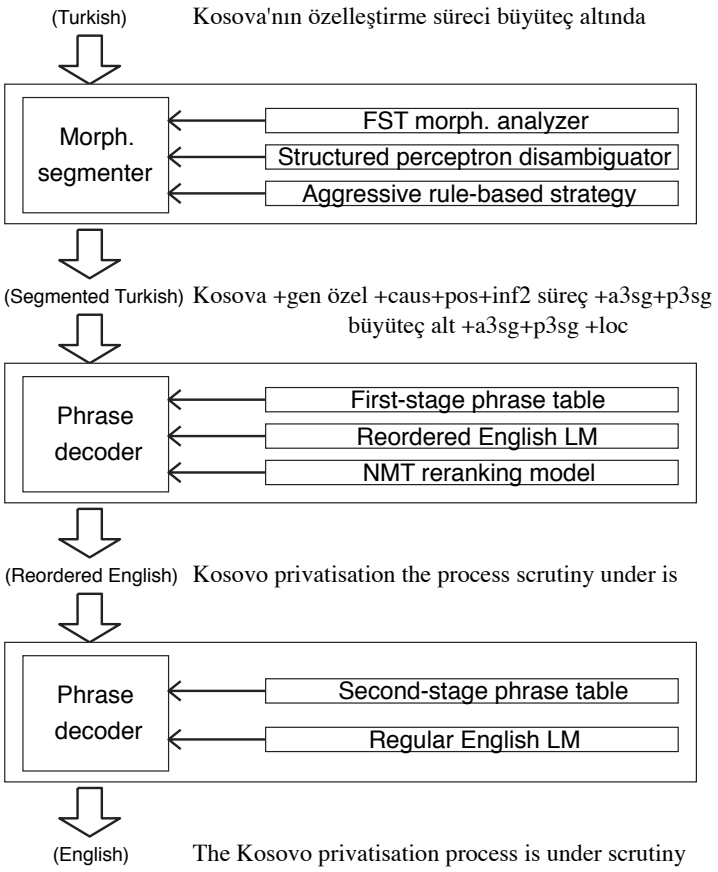


Figure 5: Pipeline of the submitted Turkish-English system

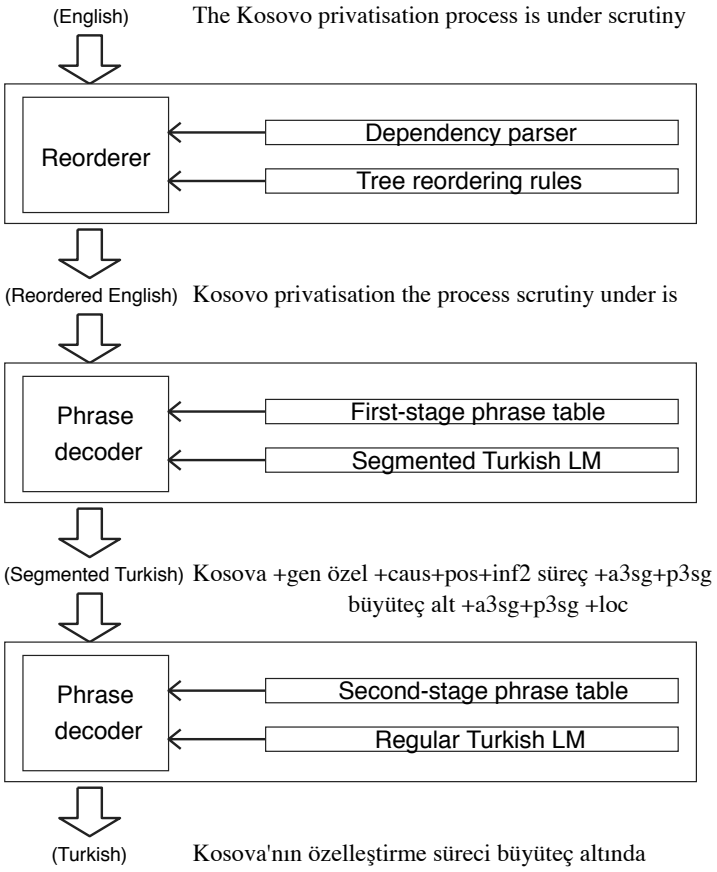


Figure 6: Pipeline of the submitted English-Turkish system