Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies

Deniz Yuret
Koç University
34450 Sariyer, Istanbul, Turkey
dyuret@ku.edu.tr

Ergun Biçici
Koç University
34450 Sariyer, Istanbul, Turkey
ebicici@ku.edu.tr

Abstract

We experiment with splitting words into their stem and suffix components for modeling morphologically rich languages. We show that using a morphological analyzer and disambiguator results in a significant perplexity reduction in Turkish. We present flexible n-gram models, FlexGrams, which assume that the $n-1$ tokens that determine the probability of a given token can be chosen anywhere in the sentence rather than the preceding $n-1$ positions. Our final model achieves 27% perplexity reduction compared to the standard n-gram model.

1 Introduction

Language models, i.e. models that assign probabilities to sequences of words, have been proven useful in a variety of applications including speech recognition and machine translation (Bahl et al., 1983; Brown et al., 1990). More recently, good results on lexical substitution and word sense disambiguation using language models have also been reported (Hawker, 2007; Yuret, 2007). Morphologically rich languages pose a challenge to standard modeling techniques because of their relatively large out-of-vocabulary rates and the regularities they possess at the sub-word level.

The standard n-gram language model ignores long-distance relationships between words and uses the independence assumption of a Markov chain of order $n-1$. Morphemes play an important role in the syntactic dependency structure in morphologically rich languages. The dependencies are not only between stems but also between stems and suffixes and if we use complete words as unit tokens, we will not be able to represent these sub-word dependencies. Our working hypothesis is that the performance of a language model is correlated by how much the probabilistic dependencies mirror the syntactic dependencies. We present flexible n-grams, FlexGrams, in which each token can be conditioned on tokens anywhere in the sentence, not just the preceding $n-1$ tokens. We also experiment with words split into their stem and suffix forms, and define stem-suffix FlexGrams where one set of offsets is applied to stems and another to suffixes. We evaluate the performance of these models on a morphologically rich language, Turkish.

2 The FlexGram Model

The FlexGram model relaxes the contextual assumption of n-grams and assumes that the $n-1$ tokens that determine the probability of a given token can be chosen anywhere in the sentence rather than at the preceding $n-1$ positions. This allows the ability to model long-distance relationships between tokens without a predefined left-to-right ordering and opens the possibility of using different dependency patterns for different token types.

Formal definition An order-n FlexGram model is specified by a tuple of dependency offsets $[d_1, d_2, \ldots, d_{n-1}]$ and decomposes the probability of a given sequence of tokens into a product of conditional probabilities for every token:

$$p(w_1, \ldots, w_k) = \prod_{w_i \in S} p(w_i | w_{i+d_1} \ldots w_{i+d_{n-1}})$$

The offsets can be positive or negative and the same set of offsets is applied to all tokens in the sequence. In order to represent a properly normalized probability model over the set of all finite length sequences, we check that the offsets of a FlexGram model does not result in a cycle. We show that using differing dependency offsets for stems and suffixes can improve the perplexity.

3 Dataset

We used the Turkish newspaper corpus of Milliyet after removing sentences with 100 or more tokens. The dataset contains about 600 thousand sentences in the training set and 60 thousand sentences in the test set (giving a total of about 10 million words). The versions of the corpus we use developed by using different word-split strategies along with a sample sentence are explained below:

1. The *unsplit* dataset contains the raw corpus:
   `Kasparov bükemediği eli öpecek`
   (Kasparov is going to kiss the hand he cannot bend)
2. The *morfessor* dataset was prepared using the Morfessor (Creutz et al., 2007) algorithm:
   `Kasparov büke +mediği eli öp +ecek`
3. The *auto-split* dataset is obtained after using our unsupervised morphological splitter:
   `Kaspar +ov bük +emediği eli öp +ecek`
4. The *split* dataset contains words that are split into their stem and suffix forms by using a highly accurate supervised morphological analyzer (Yuret and Türe, 2006):
   `Kasparov bük +yAmA+dHk+sH el +sH öp +yAcAk`
5. The *split+0* version is derived from the *split* dataset by adding a zero-suffix to any stem that is not followed by a suffix:
   `Kasparov +0 bük +yAmA+dHk+sH el +sH öp +yAcAk`

Some statistics of the dataset are presented in Table 1. The vocabulary is taken to be the tokens that occur more than once in the training set and the OOV column shows the number of out-of-vocabulary tokens in the test set. The unique and 1-count columns give the number of unique tokens and the number of tokens that only occur once in the training set. Approximately 5% of the tokens in the *unsplit* test set are OOV tokens. In comparison, the ratio for a comparably sized English dataset is around 1%. Splitting the words into stems and suffixes brings the OOV ratio closer to that of English.

Model evaluation When comparing language models that tokenize data differently:

1. We take into account the true cost of the OOV tokens using a separate character-based model similar to Brown et al. (1992).
2. When reporting averages (perplexity, bits-per-word) we use a common denominator: the number of unsplit words.

Table 1: Dataset statistics (K for thousands, M for millions)

Dataset	Train	Test	OOV	Unique	1-count
unsplit	8.88M	0.91M	44.8K (4.94%)	430K	206K
morfessor	9.45M	0.98M	10.3K (1.05%)	167K	34.4K
auto-split	14.3M	1.46M	13.0K (0.89%)	128K	44.8K
split	12.8M	1.31M	17.1K (1.31%)	152K	75.4K
split+0	17.8M	1.81M	17.1K (0.94%)	152K	75.4K

4 Experiments

In this section we present a number of experiments that demonstrate that when modeling a morphologically rich language like Turkish, (i) splitting words into their stem and suffix forms is beneficial when the split is performed using a morphological analyzer and (ii) allowing the model to choose stem and suffix dependencies separately and flexibly results in a perplexity reduction, however the reduction does not offset the cost of zero suffixes. We used the SRILM toolkit (Stolcke, 2002) to simulate the behavior of FlexGram models by using count files as input. The interpolated Kneser-Ney smoothing was used in all our experiments.

Table 2: Total log probability (M for millions of bits).

	Split Dataset		Unsplit Dataset	
N	Word logp	OOV logp	Word logp	OOV logp
1	14.2M	0.81M	11.7M	2.32M
2	10.5M	0.64M	9.64M	1.85M
3	9.79M	0.56M	9.46M	1.59M
4	9.72M	0.53M	9.45M	1.38M
5	9.71M	0.51M	9.45M	1.25M
6	9.71M	0.50M	9.45M	1.19M

4.1 Using a morphological tagger and disambiguator

The *split* version of the corpus contains words that are split into their stem and suffix forms by using a previously developed morphological analyzer (Oflazer, 1994) and morphological disambiguator (Yuret and Türe, 2006). The analyzer produces all possible parses of a Turkish word using the two-level morphological paradigm and the disambiguator chooses the best parse based on the analysis of the context using decision lists. The integrated system was found to discover the correct morphological analysis for 96% of the words on a hand annotated out-of-sample test set. Table 2 gives the total log-probability (using $\log_2$) for the split and unsplit datasets using n-gram models of different order. We compute the perplexity of the two datasets using a common denominator: $2^{-\log_2(p)/N}$ where N=906,172 is taken to be the number of unsplit tokens. The best combination (order-6 word model combined with an order-9 letter model) gives a perplexity of 2,465 for the split dataset and 3,397 for the unsplit dataset, which corresponds to a 27% improvement.

4.2 Separation of stem and suffix models

Only 45% of the words in the *split* dataset have suffixes. Each sentence in the *split+0* dataset has a regular *[stem suffix stem suffix ...]* structure. Table 3 gives the average cost of stems and suffixes in the two datasets for a regular 6-gram word model (ignoring the common OOV words). The log-probability spent on the zero suffixes in the split+0 dataset has to be spent on trying to decide whether to include a stem or suffix following a stem in the split dataset. As a result the difference in total log-probability between the two datasets is small (only 6% perplexity difference). The set of OOV tokens is the same for both the *split* and *split+0* datasets; therefore we ignore the cost of the OOV tokens as is the default SRILM behavior.

Table 3: Total log probability for the 6-gram word models on split and split+0 data.

	split dataset		split+0 dataset	
token type	number of tokens	total $-\log_2 p$	number of tokens	total $-\log_2 p$
stem	0.91M	7.80M	0.91M	7.72M
suffix	0.41M	1.89M	0.41M	1.84M
0-suffix	–	–	0.50M	0.21M
all	1.31M	9.69M	1.81M	9.78M

4.3 Using the FlexGram model

We perform a search over the space of dependency offsets using the *split+0* dataset and considered n-gram orders 2 to 6 and picked the dependency offsets within a window of $4n + 1$ tokens centered around the target. Table 4 gives the best models discovered for stems and suffixes separately and compares them to the corresponding regular n-gram models on the split+0 dataset. The numbers in parentheses give perplexity and significant reductions can be observed for each n-gram order.

Table 4: Regular ngram vs FlexGram models.

N	ngram-stem	ngram-suffix
2	-1 (1252)	-1 (5.69)
3	-2,-1 (418)	-2,-1 (5.29)
4	-3,-2,-1 (409)	-3,-2,-1 (4.79)
5	-4,-3,-2,-1 (365)	-4,-3,-2,-1 (4.80)
6	-5,-4,-3,-2,-1 (367)	-5,-4,-3,-2,-1 (4.79)

N	flexgram-stem	flexgram-suffix
2	-2 (596)	-1 (5.69)
3	+1,-2 (289)	+1,-1 (4.21)
4	+2,+1,-1 (189)	-2,+1,-1 (4.19)
5	+4,+2,+1,-1 (176)	-3,-2,+1,-1 (4.12)
6	+4,+3,+2,+1,-1 (172)	-4,-3,-2,+1,-1 (4.13)

However, some of these models cannot be used in combination because of cycles as we depict on the left side of Figure 1 for order 3. Table 5 gives the best combined models without cycles. We were able to exhaustively search all the patterns for orders 2 to 4 and we used beam search for orders 5 and 6. Each model is represented by its offset tuple and the resulting perplexity is given in parentheses. Compared to the regular n-gram models from Table 4 we see significant perplexity reductions up to order 4. The best order-3 stem-suffix FlexGram model can be seen on the right side of Figure 1.

Table 5: Best stem-suffix flexgram model combinations for the split+0 dataset.

N	flexgram-stem	flexgram-suffix	perplexity reduction
2	-2 (596)	-1 (5.69)	52.3%
3	-4,-2 (496)	+1,-1 (4.21)	5.58%
4	-4,-2,-1 (363)	-3,-2,-1 (4.79)	11.3%
5	-6,-4,-2,-1 (361)	-3,-2,-1 (4.79)	1.29%
6	-6,-4,-2,-1 (361)	-3,-2,-1 (4.79)	1.52%

5 Related work

Several approaches attempt to relax the rigid ordering enforced by the standard n-gram model. The skip-gram model (Siu and Ostendorf, Jan 2000) allows the skipping of one word within a given n-gram. Variable context length language modeling (Kneser, 1996) achieves a 10% perplexity reduction when compared to the trigrams by varying the order of the n-gram model based on the context. Dependency models (Rosenfeld, 2000) use the parsed dependency structure of sentences to build the language model as in grammatical trigrams (Lafferty et al., 1992), structured language models (Chelba and Jelinek, 2000), and dependency language models (Chelba et al., 1997). The dependency model governs the whole sentence and each word in a sentence is likely to have a different dependency structure whereas in our experiments with FlexGrams we use two connectivity patterns: one for stems and one for suffixes without the need for parsing.

6 Contributions

We have analyzed the effect of word splitting and unstructured dependencies on modeling Turkish, a morphologically complex language. Table 6 compares the models we have tested on our test corpus.

We find that splitting words into their stem and suffix components using a morphological analyzer and disambiguator results in significant perplexity reductions of up to 27%. FlexGram models outperform regular n-gram models (Tables 4 and 5)

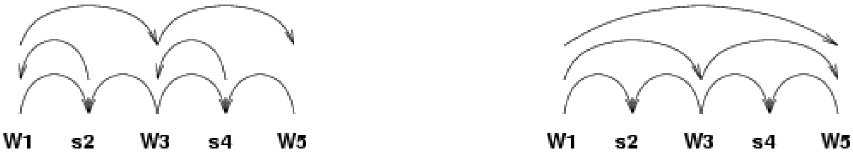

Figure 1: Two FlexGram models where W represents a stem, s represents a suffix, and the arrows represent dependencies. The left model has stem offsets [+1,-2] and suffix offsets [+1,-1] and cannot be used as a directed graphical model because of the cycles. The right model has stem offsets [-4,-2] and suffix offsets [+1,-1] and is the best order-3 FlexGram model for Turkish.

Table 6: Perplexity for compared models.

N	unsplit	split	flexgram
2	3929	4360	5043
3	3421	2610	3083
4	3397	2487	2557
5	3397	2468	2539
6	3397	2465	2539

when using an alternating stem-suffix representation of the sentences; however Table 6 shows that the cost of the alternating stem-suffix representation (zero-suffixes) offsets this gain.

References

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

Peter F. Brown et al. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1992.

Ciprian Chelba and Frederick Jelinek. Recognition performance of a structured language model. *CoRR*, cs.CL/0001022, 2000.

Ciprian Chelba, David Engle, Frederick Jelinek, Victor M. Jimenez, Sanjeev Khudanpur, Lidia Mangu, Harry Printz, Eric Ristad, Ronald Rosenfeld, Andreas Stolcke, and Dekai Wu. Structure and performance of a dependency language model. In *Proc. Eurospeech '97*, pages 2775–2778, Rhodes, Greece, September 1997.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraclar, and Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *TSLP*, 5 (1), 2007.

Tobias Hawker. USYD: WSD and lexical substitution using the Web1T corpus. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, 2007.

R. Kneser. Statistical language modeling using a variable context length. In *Proc. ICSLP '96*, volume 1, pages 494–497, Philadelphia, PA, October 1996.

John Lafferty, Daniel Sleator, and Davy Temperley. Grammatical trigrams: a probabilistic model of link grammar. In *AAAI Fall Symposium on Probabilistic Approaches to NLP*, 1992.

Kemal Oflazer. Two-level description of turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148, 1994.

Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, volume 88, pages 1270–1278, 2000.

Manhung Siu and M. Ostendorf. Variable n-grams and extensions for conversational speech language modeling. *Speech and Audio Processing, IEEE Transactions on*, 8(1):63–75, Jan 2000. ISSN 1063-6676. doi: 10.1109/89.817454.

Andreas Stolcke. Srilm – an extensible language modeling toolkit. In *Proc. Int. Conf. Spoken Language Processing (ICSLP 2002)*, 2002.

Deniz Yuret. KU: Word sense disambiguation by substitution. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, June 2007.

Deniz Yuret and Ferhan Türe. Learning morphological disambiguation rules for turkish. In *HLT-NAACL 06*, June 2006.