A Toolkit for Efficient Learning of Lexical Units for Speech Recognition

Matti Varjokallio, Mikko Kurimo
Aalto University, Department of Signal Processing and Acoustics
Espoo, Finland
matti.varjokallio@aalto.fi, mikko.kurimo@aalto.fi

Abstract

String segmentation is an important and recurring problem in natural language processing and other domains. For morphologically rich languages, the amount of different word forms caused by morphological processes like agglutination, compounding and inflection, may be huge and causes problems for traditional word-based language modeling approach. Segmenting text into better modelable units is thus an important part of the modeling task. This work presents methods and a toolkit for learning segmentation models from text. The methods may be applied to lexical unit selection for speech recognition and also other segmentation tasks.

Keywords: Language modeling, Lexical units, Automatic speech recognition

1. Introduction

Many natural language processing tasks including automatic speech recognition are relying on the statistical n-gram language modeling paradigm (Manning and Schütze, 1999), in which the probability of a string is conditioned on the (n-1) predecessor strings:

$$P(w_n) \approx P(w_n|w_1^{n-1}) \tag{1}$$

Traditionally the n-gram model is estimated over words. For morphologically rich languages, the amount of different word forms caused by morphological processes like agglutination, compounding and inflection, may be huge. In this case, estimating the n-gram model over sequences of words is likely to suffer from high Out-Of-Vocabulary (OOV) -rate and unreliable n-gram estimates as a result of data sparsity. Segmentation of text into better modelable units thus becomes an important part of the modeling task. OOV -issues are avoided altogether, as all word forms may be generated by concatenating the base units.

Many possible objectives are available for the segmentation task. Smallest meaning-bearing units, morphs, are a viable linguistically motivated target for segmentation. Different machine learning approaches have been suggested in the literature. Statistical segmentations have been evaluated in the Morpho Challenge competitions (Kurimo et al., 2010) and have been shown to perform well across various benchmark tasks in automatic speech recognition, machine translation and information retrieval. (Hirsimäki et al., 2009) provides a survey to speech recognition results on many languages. The advantage of unsupervised methods lies in that neither a morphological analyzer nor an annotated training corpus is required.

The viewpoint in this work is closely related to the n-gram language modeling approach and their application to automatic speech recognition. For a speech recognition task, a large high-order n-gram model is trained from a large text corpus. The goodness of the model is then evaluated by how well it predicts text by measures such as cross-entropy or perplexity (Goodman, 2001). As selecting high-order n-grams for unknown segmentation is infeasible, we attempt to learn lower-order segmentation models which predict the obtained unit sequence with a high likelihood.

We present a toolkit for learning unigram and bigram segmentation models from text. The method has so far been successfully applied to segmenting words into subword units for language modeling in large vocabulary speech recognition (Varjokallio et al., 2013) and was shown to result in an efficient vocabulary for the task. Here it is also extended to infer phrase-like segments from sentences on large corpora and to utilize bigram statistics in subword segmentation.

2. Segmentation models

Generating text using a vocabulary with an n-gram distribution over the vocabulary units may be viewed as an (n-1)-order Markov process. With respect to inferring model parameters from unsegmented text, the states of the process are not directly observable, because the states emit strings of varying length and the borders between the emitted strings are not observed. Parameter inference in the unigram case has been addressed in the multigram framework (Deligne and Bimbot, 1997). Bigram statistics over class information were utilized in (Deligne and Sagisaka, 1998). In practice, Expectation-Maximization -training (Dempster et al., 1977) with the Forward-backward algorithm and the Viterbi approximation may be applied.

Model selection for a vocabulary of limited size is a non-trivial task. In the general sense, searching for a vocabulary with evenly distributed frequencies is known to be a NP-complete problem (Storer, 1988). For natural language data, it could be expected that reasonable approximations may be found. For both unigram and bigram statistics, we employ a likelihood based pruning scheme. The approach taken is to start with a large vocabulary, which is then pruned to a suitable size. The training proceeds in a greedy fashion, i.e. in each iteration, strings, which are the least significant for the data likelihood, are removed from the vocabulary. This has experimentally given good results for subword and phrase segmentation. The type of the units is selected in the initialization phase.

2.1. Unigram model

The algorithm aims to learn a vocabulary that gives a high unigram likelihood for the training corpus. The vocabulary V consists of substrings s_i and a probability lp_i for each substring. The training corpus C consists of strings weighted by their frequency in the corpus.

Figure 1 shows an example how Finnish word “talossa” could be segmented as a sequence of letters, subwords, or as a single observation with unigram scores. The most likely segmentation returned by the Viterbi algorithm would in this case be “talo + ssa”. The graph structure does not need to be explicitly constructed. For efficient subword lookups starting from each character position, the vocabulary is stored in a letter-trie data structure.

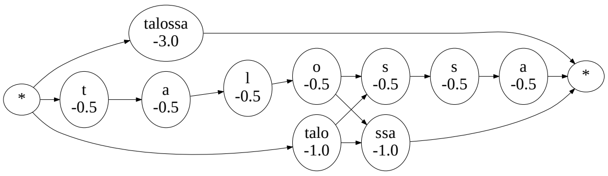

Figure 1: Segmentation paths for word “talossa” using a subword unigram model. The numbers are log-likelihoods of the units.

The algorithm proceeds by starting with a large vocabulary, which is then pruned to a suitable size. In the current approach, no new strings are introduced, and thus proper initialization is important.

Initialization

1. Train a letter n-gram model from the training corpus.
2. Select the initial pool of strings $V = \{s_i\}$, for example all substrings from the most common words in the training data up to a reasonable maximum length.
3. Calculate an initial log-probability lp_i for each string s_i using the letter n-gram model:

$$lp_i(abcd) = lp(a)+lp(b|a)+lp(c|ab)+lp(d|abc) \quad (2)$$

Normalize the probabilities to sum to one. Zerogram initialization is also plausible if Forward-backward is iterated in the next step.

4. Iterate Forward-backward over the training corpus until convergence.
5. Iterate training. After each iteration increase cutoff value and remove strings with frequency below the cutoff value.

Vocabulary pruning

The pruning approach tries to account for the effect that removing a subword has on the likelihood. Iterate:

1. Resegment the training data and update string probabilities.
2. Select a list of candidate strings for removal, for example the least frequent strings in the vocabulary.
3. For each candidate string, estimate the cost of removing it by resegmenting the training data without it.
4. Sort the list of candidate strings in descending order by the value of estimated likelihood change.
5. Remove a defined amount of top candidate strings. Alternatively it is possible to update parameters after each removal and verify that the cost for each subsequent removal is above a threshold value.

Iteration may be stopped when the desired vocabulary size is reached.

2.2. Bigram model

Analogously to the unigram model, generating text using a vocabulary with bigram dependencies over the vocabulary units may be viewed as a first-order Markov process. The model consists of a vocabulary V of substrings $s_i \in V$ and bigram probabilities $lp(s_i|s_j)$ for $s_i \in V$ and $s_j \in V$. The training corpus C consists of strings weighted by their frequency in the corpus. When training a bigram model in Expectation Maximization-style, special attention is needed in selecting the data structures, as all bigram transitions in the corpus need to be represented. A graph containing all segmentations may be constructed separately for each word in the training data. Figure 2 contains segmentations and corresponding bigram probabilities for a single word. For most cases, it is more efficient to merge all graphs to a joint graph. Figure 3 presents possible segmentations for three Finnish words, “talo”, “talossa” and “talous” in a joint graph.

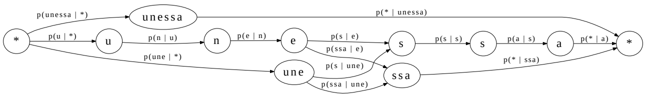

Figure 2: Segmentation paths for Finnish word “unessa” in a separate graph using a subword bigram model.

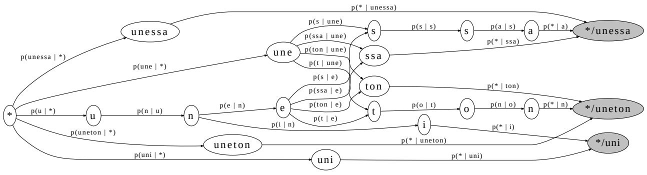

Figure 3: Segmentation paths for three Finnish words: “uni”, “unessa” and “uneton” in a joint graph using a subword bigram model.

Each arc is assigned a pointer to the corresponding bigram score. The prefixes are shared for each word and a unique end node is assigned for each word. If word boundaries are modeled as a separate symbol, the EM-training procedure is optimizing the bigram likelihood of the whole corpus for the parameter set.

In the joint graph structure, it is efficient to train Forward-backward for all the words simultaneously. Forward pass may be done for the whole graph in a single pass for all words. The backward pass is done for each word separately starting from the end node of the word. When computing likelihood for a single word, the forward pass may equivalently be done in a backward direction, starting from the end node.

The vocabulary pruning may be done in a similar fashion as with the unigram statistics. In practice, the vocabulary can be pruned to a reasonable size with unigram statistics and proceeding with bigram statistics for the rest of the training. Model selection with bigram statistics is more complicated than in the unigram case, as it is possible to prune both the vocabulary and bigrams.

3. Discussion

For morphologically rich languages, words are troublesome as units for natural language processing tasks, because of Out-Of-Vocabulary and data sparsity issues. Segmenting text to better modelable units for language model training helps in solving these issues.

Due to the present-day availability of large text corpora from internet sources, there is also interest in using large vocabularies for languages, which may not necessarily be classified as morphologically rich. Text quality may also vary because of misspellings etc. We believe that language modeling by units obtained by resegmenting the corpus may help in this endeavour. Advantages are that the models will be OOV-free and also all the training data may be utilized in the training phase.

The unigram segmentation model has been used so far for subword segmentation and phrase segmentation. The type of the subword units may be controlled by either training the model with word types or word tokens. Trained with word types, the selected units resemble morphs. Training with word tokens attempts to minimize the unigram entropy, which may be a good property for statistical NLP tasks. In (Varjokallio et al., 2013) this was shown to result in an efficient vocabulary for Finnish large vocabulary speech recognition.

Training a phrase segmentation model including words, multiwords, subwords and cross-word segments is possible on a large corpus. This type of segmentation model can be useful for NLP tasks in languages with rich morphology and also including phenomena that span multiple words. Table 1 shows example sentences segmented with a phrase model. This model was initialized from substrings of words and most common bigrams and pruned from the initial size of $8M$ strings to a size of $38k$ strings. Word boundaries are in this example modeled as part of the prefixes. The phrase segmentation model may be viewed as a statistical implementation of a linear unit grammar (Sinclair and Mauranen, 2006).

Perhaps the most interesting units in the example are the suffix + word style dependencies: *”ksi_valittiin” ~ ”was selected/appointed as”* and *”jen_perusteella” ~ ”based on/PLURAL”*. The training complexity may be controlled by initialization and pruning parameters. The model in the example was trained on 150 million word tokens, and the training time was one week with a single core implementation.

Table 2 contains preliminary word perplexity results for n-gram models trained over varying types of unigram segmentations and training parameters. The models were trained on the Finnish Kielipankki corpus (CSC - Scientific Computing Ltd., 2003) which contains around $150M$ word tokens and $4.1M$ word types and evaluated on a held-out set of $3.9M$ word tokens. The training corpus was segmented with the corresponding segmentation model and the final modified Kneser-Ney smoothed n-gram model was trained using the VariKN language modeling toolkit (Siivola et al., 2007). Model sizes were controlled by the pruning parameters. For the segmentation models, where the word boundary was modeled as part of the units, the training was done by appending the word boundary symbol to the left side of each word and disallowing a separate word boundary symbol. Thus, the subword vocabularies for ”Symbol” and ”Left” cases are different. For earlier perplexity results in Finnish and a parallel study with English, see (Siivola et al., 2007).

Table 1: Example Finnish sentence segmentations with a phrase model. Word boundaries are marked with the ”_” sign.

- _joulun _hitti tuote tta _ei_kukaan _halua _vielä _tässä_vaiheessa _veikka illa
- _ilomantsin _tunnus laulu ksi_valittiin _aulis _raita lan _koi tere _laulu
- _tällaiseen kin _johto päätökseen _on _aihetta _mielipidetiedustelu jen_perusteella
- _kokemukset _ovat_olleet _hyvät

Model	WB	Cutoff	Order	Size	Perpl.
Subword	Symbol	222222	6	16.0M	5440
Subword	Left	222111	6	16.0M	5255
Subword	Left	222111	6	21.6M	4768
Subword	Left	22111	5	31.0M	4303
Phrase	Left	221	3	24.5M	5087
Phrase	Left	2211	4	32.8M	4488

Table 2: Preliminary Finnish n-gram word perplexities for different model types segmented with a unigram segmentation model. “WB” stands for the type of word boundary modeling. Lexicon size is $35k$ for subword models and $38k$ for phrase models. Size is the number of n-grams in the model.

It seems that for a comparable model size slightly better perplexity values are reached by modeling word boundaries as part of the first subword of the word instead of a separate word boundary symbol. The most accurate model was a subword model trained with word boundary on left. Phrase models did not at the moment improve perplexities compared to subword models. However, unigram, bigram and trigram distributions over phrase segmentations are more efficient than for subwords. A phrase-segmented model may thus be a worthwhile choice for some speech recognition tasks.

The unigram segmentation algorithm may also be viewed as a general purpose Markov-0 compressor. As the approach scales to long strings, other possible uses could include offline data compression and string processing in bioinformatics.

Training a bigram segmentation model is currently limited to subword segmentation (on a latin-style alphabet). Bigram statistics have been applied at least for Chinese word segmentation (Goldwater et al., 2006). Model selection for bigram segmentations is currently more experimental.

In model selection the methods rely on purely likelihood-based pruning and user-controlled vocabulary size. Information theorical penalized criteria are currently a more common approach to the model selection problem. One existing method is Morfessor (baseline) (Creutz and Lagus, 2002), which optimizes a criterion derived from the Minimum Description Length (MDL) -principle. The approach in the present work expects that a reasonably large corpus is available and an n-gram model will be trained over the corpus. In this case, cost function for the vocabulary coding will be insignificant as the number of n-grams will dominate the n-gram model complexity.

Proper pronunciation modeling is important for a speech recognition task. The suggested pruning approach is well suited for creating recognition vocabularies also for languages with more complex letter-to-phoneme -mapping. The vocabulary may be initialized by selecting only strings with a well defined pronunciation variant. Word boundary modeling is also something to consider both with respect to n-gram training and decoding in the recognition phase. As seen in table 2 the word boundary modeling has an effect on the perplexities. Also the decoding graph needs to be constructed differently for each of the unit types for correct decoding.

4. Software

The toolkit aims to be a focused contribution for studying n-gram training and model selection in the case of unknown segmentation. The toolkit has been implemented in `C++` programming language and is available as BSD-3 -licenced open source from the address `http://www.github.com/aalto-speech/ftk`. It includes functionality for segmentation, EM-training and model selection. Multiple implementations of Forward-backward and Viterbi algorithms for both unigram and bigram statistics are included. The model selection and pruning functionality has been implemented as a separate abstraction level, and is extendable for different purposes. Most important parts of the toolkit are unit tested.

5. Conclusion

We have presented a toolkit for efficiently learning unigram and bigram segmentation models from text. The methods may be applied to lexical unit selection for speech recognition and also other segmentation tasks.

6. Acknowledgements

This work was financially supported by the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant no. 251170) and LASTU Programme (nos. 256887 and 259934). Computational resources were provided by the Aalto Science-IT project.

7. References

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30, Philadelphia, Pennsylvania, USA.

CSC - Scientific Computing Ltd. (2003). Kielipankki corpus. An electronic document collection of the Finnish language.

Deligne, S. and Bimbot, F. (1997). Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23(3):223–241.

Deligne, S. and Sagisaka, Y. (1998). Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. In *COLING-ACL*, pages 300–306.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

Goodman, J. T. (2001). A bit of progress in language modeling. Extended Version. Technical report, Microsoft Research.

Hirsimäki, T., Pylkkönen, J., and Kurimo, M. (2009). Importance of high-order N-gram models in morph-based speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 17(4):724–732.

Kurimo, M., Virpioja, S., Turunen, V., and Lagus, K. (2010). Morpho challenge competition 2005–2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, pages 87–95, Stroudsburg, PA, USA. Association for Computational Linguistics.

Manning, C. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA.

Siivola, V., Hirsimäki, T., and Virpioja, S. (2007). On growing and pruning Kneser-Ney smoothed N-gram models. *IEEE Transactions on Speech, Audio and Language Processing*, 15(5):1617–1624.

Sinclair, J. and Mauranen, A. (2006). *Linear unit grammar: Integrating speech and writing (Vol. 25). Amsterdam: John Benjamins.*

Storer, J. A. (1988). *Data Compression: Methods and Theory*. Computer Science Press, a subsidiary of W. H. Freeman & Company.

Varjokallio, M., Kurimo, M., and Virpioja, S. (2013). Learning a subword vocabulary based on unigram likelihood. In *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Olomouc, Czech Republic.