An Improved Hierarchical Word Sequence Language Model Using **Directional Information**

Xiaoyi Wu

Computational Linguistics Laboratory 8916-5 Takayama, Ikoma, Nara Japan xiaoyi-w@is.naist.jp

Abstract

For relieving data sparsity problem, Hierarchical Word Sequence (abbreviated as HWS) language model, which uses word frequency information to convert raw sentences into special n-gram sequences, can be viewed as an effective alternative to normal n-gram method. In this paper, we use directional information to make HWS models more syntactically appropriate so that higher performance can be achieved. For evaluation, we perform intrinsic and extrinsic experiments, both verify the effectiveness of our improved model.

1 Introduction

Probabilistic Language Modeling is a fundamental research direction of Natural Language Processing. It is widely used in many applications such as machine translation (Brown et al., 1990), spelling correction (Mays et al., 1991), speech recognition (Rabiner and Juang, 1993), word prediction (Bickel et al., 2005) and so on.

Most research about Probabilistic Language Modeling, such as back-off (Katz, 1987), Kneser-Ney (Kneser and Ney, 1995), and modified Kneser-Ney (Chen and Goodman, 1999), only focus on smoothing methods because they all take n-gram approach (Shannon, 1948) as a default setting for extracting word sequences from a sentence. Yet even with 30 years worth of newswire text, more than one third of all trigrams are still unseen (Allison et al., 2005), which cannot be distinguished accurately even using a high-performance smoothing method such as modified Kneser-Ney (abbreviated as MKN). It is

Yuji Matsumoto

Nara Institute of Science and Technology Nara Institute of Science and Technology **Computational Linguistics Laboratory** 8916-5 Takayama, Ikoma, Nara Japan matsu@is.naist.jp

> better to make these unseen sequences actually be observed rather than to leave them to smoothing method directly.

> For the purpose of extracting more valid word sequences and relieving data sparsity problem, Wu and Matsumoto (2014) proposed a heuristic approach to convert a sentence into a hierarchical word sequence (abbreviated as HWS) structure, by which special ngrams can be achieved. In this paper, we improve HWS models by adding directional information for achieving higher performance.

> This paper is organized as follows. In Section 2, we give a complete review of the HWS language model. We present our improved HWS model in Section 3. In Section 4, we show the effectiveness of our model by several experiments. Finally, we summarize our findings in Section 5.

Review of HWS Language Model 2

The HWS language model is defined as follows.

Suppose that we have a frequency-sorted vocabulary list $V = \{v_1, v_2, ..., v_m\}$, where $C(v_1) \ge$ $C(v_2) \ge \dots \ge C(v_m)^1.$ According to V, given any sentence $S = w_1, w_2, ..., w_n,$ the most frequently used word $w_i \in S(1 \le i \le n)$ can be selected² for splitting S into two substrings $S_L = w_1, \dots, w_{i-1}$ and $S_R = w_{i+1}, \dots, w_n$. Similarly, for S_L and S_R , $w_i \in S_L (1 \le j \le i - 1)$ and $w_k \in S_R(i+1 \le k \le n)$ can also be se-

lected, by which S_L and S_R can be splitted

 $^{{}^{1}}C(v)$ represents the frequency of v in a certain corpus.

²If w_i appears multiple times in S, then select the first one.

⁴⁴⁹

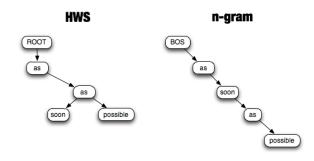


Figure 1: A comparison of structures between HWS and n-gram

into two smaller substrings separately. Executing this process recursively until all the substrings become empty strings, then a tree $T = (\{w_i, w_j, w_k, ...\}, \{(w_i, w_j), (w_i, w_k), ...\})$ can be generated, which is defined as an *HWS structure*.

In an HWS structure T, assuming that each node depends on its preceding n-1 parent nodes, then special n-grams can be trained. Such kind of n-grams are defined as *HWS-n-grams*.

The advantage of HWS models can be considered as discontinuity. Taking Figure 1 as an example, since n-gram model is a continuous language model, in its structure, the second 'as' depends on 'soon', while in the HWS structure, the second 'as' depends on the first 'as', forming a discontinuous pattern to generate the word 'soon', which is closer to our linguistic intuition. Rather than 'as soon ...', taking 'as ... as' as a pattern is more reasonable because 'soon' is quite easy to be replaced by other words, such as 'fast', 'high', 'much' and so on. Consequently, even using 4-gram or 5-gram, sequences consisting of 'soon' and its nearby words tend to be lowfrequency because the connection of 'as...as' is still interrupted. On the contrary, the HWS model extracts sequences in a discontinuous way, even 'soon' is replaced by another word, the expression 'as...as' won't be affected. This is how the HWS models relieve the data sparseness problem.

It unsupervisedly construct a hierarchical structure to adjust the word sequence so that irrelevant words can be filtered out from contexts and long distance information can be used for predicting the next word. On this point, it has something in common with structured language model (Chelba, 1997), which firstly introduced parsing into language modeling. The significant difference is, structured language model is based on CFG parsing structures, while HWS model is based on patternoriented structures.

The experimental results reported by Wu and Matsumoto (2014) indicated that HWS model keeps better balance between coverage and usage than normal n-gram and skip-gram models (Guthrie, 2006), which means that more valid sequence patterns can be extracted in this approach.

However, the *discontinuity* of HWS models also brings a disadvantage. In normal n-gram models, since the generation of words is one-sided (from left to right), given any left-hand context, words generated from it can be considered as linguistically appropriate. In contrast, HWS structures are essentially binary trees, which also generate words on the left side. However, according to the definition of HWS-n-grams, the directional information are not taken into account, which causes a syntactical problem.

Taking Figure 1 as an example. According to the structure of HWS, HWS-3-grams are trained as {(ROOT, as, as), (as, as, soon), (as, as, possible)}, where 'soon' and 'possible' are generated from context (as, as) without any distinction, which means, an illegal sentence such like 'as possible as soon' can be also generated from this HWS-3-gram model.

3 Directional HWS Models

To solve this problem, we propose to use directional information. As mentioned previously, since HWS structures are essentially binary trees, directional information has already been encoded when HWS structures are established.

Thus, after an HWS structure being constructed, directional information can be easily attached to this tree as shown in Figure 2. Then, assuming that each node depends on its n-1 preceding parent nodes with their directional information, we can train a special n-gram from this binary tree. For instance, 3-grams trained from this tree are {(ROOT-R, as-R, as), (as-R, as-L, soon), (as-R, as-R, possible)}, where syntactical information can be encoded more precisely than original HWS-3-grams. For the purpose of distinguishing our models from the original HWS mod-

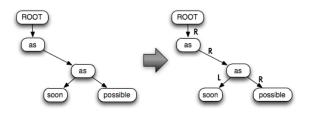


Figure 2: An example of HWS structure with directional information

els, we call n-grams trained in our way as DHWS-ngrams.

In the above example of DHWS-3-grams, (as-R, as-L, soon) indicates that 'soon' is located between two 'as's, while (as-R, as-R, possible) indicates that 'possible' is located on the right side of the second 'as'. Similarly, if we use DHWS-4-grams or higher order ones, the relative position of each word will be more specific. In other words, according to a DHWS structure, for each word (node), its position (relative to the whole sentence) can be strictly determined by its preceding parent nodes. The bigger n is, the more syntactical information DHWS-n-grams can reflect.

As for smoothing methods for HWS models, Wu and Matsumoto (2014) only used an additive smoothing. Although HWS-n-grams are trained in a special way, they are essentially n-grams because each trained sequence is reserved as a $(n - 1 \ length \ context, \ word)$ tuple as normal n-grams, which makes it possible to apply MKN smoothing to HWS models. The main difference is that HWS models are trained by tree structures while n-gram models in a continuous way, which affects the counting of contexts $C(w_{i-n+1}^{i-1})$.

Taking Figure 1 as an example. According to the structure of HWS, HWS-3-grams are trained as {(ROOT, as, as), (as, as, soon), (as, as, possible)}, while the HWS-2-grams are trained as {(ROOT, as), (as, as), (as, soon), (as, possible)}. In the HWS-3-gram model, as the context of 'soon' and 'possible', 'as ... as' appears twice, however, in the HWS-2-gram model, C(as, as) is counted only once. In normal n-gram models, $C(w_{i-n+1}^{i-1})$ can be directly achieved from its lower model because they are continuous, but in HWS models, $C(w_{i-n+1}^{i-1})$ should be counted as $\sum_{w_j \in \{w_i: C(w_{i-n+1}^i) > 0\}} C(w_{i-n+1}^{i-1}, w_j)$, which means that the frequencies of contexts should

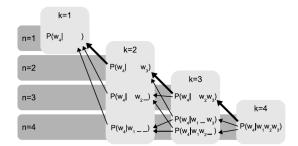


Figure 3: The interpolation of GLM model

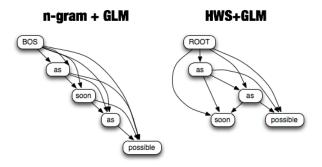


Figure 4: A demonstration for applying GLM smoothing to HWS structure

be counted in the model with the same order. Taking this into account, MKN smoothing method can be also applied to HWS models and DHWS models.

As an alternative of MKN smoothing method, we can also use GLM (Pickhardt et. al., 2014). GLM (Generalized Language Model) is a combination of skipped n-grams and MKN, which performs well on overcoming data sparseness. GLM smoothing considers all possible combinations of gaps in a local context and interpolates the higher order model with all possible lower order models derived from adding gaps in all different ways. As shown in Figure 3, n stands for the length of normal n-grams for calculation, k indicates the number of words actually be used, and the wildcard '_' represents the skipped words in a n-gram.

Since GLM is a generalized version of MKN smoothing, it can also be applied to HWS models (as shown in Figure 4). In the following experiments, we will use MKN and GLM as smoothing methods. To ensure the openness of our research, the source code used for following experiments can be downloaded.³

³https://github.com/aisophie/HWS

4 Evaluation

4.1 Intrinsic Evaluation

To test the performance on out-of-domain data, we use two different corpus: British National Corpus and English Gigaword Corpus.

British National Corpus (BNC)⁴ is a 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English from the later part of the 20th century, both spoken and written. In our experiments, we randomly choose 449,755 sentences (10 million words) as training data.

English Gigaword Corpus⁵ consists of over 1.7 billion words of English newswire from 4 distinct international sources. We randomly choose 44,702 sentences (1 million words) as test data.

As preprocessing of training data and test data, we use the tokenizer of NLTK (Natural Language Toolkit) 6 to split raw English sentences into words. We also converted all words to lowercase.

As intrinsic evaluation of Language Modeling, *perplexity* (Manning and Schütze, 1999) is the most common metric used for measuring the usefulness of a language model.

Wu and Matsumoto (2014) also proposed to use *coverage* and *usage* to evaluate efficiency of language models. The authors defined the sequences of training data as TR, and unique sequences of test data as TE, then the coverage is calculated by Equation 1.

$$coverage = \frac{|TR \bigcap TE|}{|TE|} \tag{1}$$

Usage (Equation 2) is used to estimate how much redundancy contained in a model and a balanced measure is calculated by Equation 3.

$$usage = \frac{|TR \cap TE|}{|TR|} \tag{2}$$

$$F-Score = \frac{2 \times coverage \times usage}{coverage + usage}$$
(3)

Models	PP(MKN)	PP(GLM)	С	U	F
2-gram	1244.535	-	0.479	0.081	0.139
HWS-2	1130.790	-	0.455	0.078	0.133
DHWS-2	920.783	-	0.447	0.075	0.129
3-gram	1107.430	925.666	0.229	0.028	0.051
HWS-3	1065.594	873.252	0.316	0.045	0.079
DHWS-3	834.680	687.605	0.298	0.041	0.072
4-gram	1093.799	861.930	0.086	0.009	0.016
HWS-4	1064.444	756.100	0.240	0.030	0.054
DHWS-4	822.225	596.369	0.216	0.027	0.048

Table 1: Performance of normal n-gram models, HWS models and DHWS models

Based on above measures, we compared our models with normal n-gram models and the original HWS models. The results are shown in Table 1.

According to this table, for each language model, higher order one brings lower perplexity. Besides, contrast to the result reported by Wu and Matsumoto (2014), after applied with MKN smoothing method, even for higher order models such as 3-grams and 4-grams, HWS models outperform normal n-gram models as well. Furthermore, after taking directional information into account, DHWS models perform even better than the original HWS models.

On the other hand, in DHWS models, since almost each word is distinguished as 'two words' ('-L' and '-R'), the coverage and usage tend to be relatively lower than the original HWS models. But it is worth because perplexity has been greatly decreased and syntactical information can be reflected better in this way.

We also noticed that for each model (n>2), perplexity is greatly reduced after applying GLM smoothing, which is consistent with the results reported by Pickhardt et. al.(2014).

4.2 Extrinsic Evaluation

Perplexity is not a definite way of determining the usefulness of a language model since a language model with low perplexity may not work equally well in a real world application. Thus, we also performed extrinsic experiments to evaluate our model. In this paper, we use the reranking of n-best translation candidates to examine how language models work in a statistical machine translation task.

We use the French-English part of TED talks parallel corpus as the experiment dataset. The training data contains 139761 sentence pairs, while the test

⁴http://www.natcorp.ox.ac.uk

⁵https://catalog.ldc.upenn.edu/LDC2011T07

⁶http://www.nltk.org

data contains 1617 sentence pairs. For training language models, we set English as the target language.

As for statistical machine translation toolkit, we use Moses system⁷ to train the translation model and output 50-best translation candidates for each french sentence of the test data. Then we use the 139761 English sentences to train language models. With these models, 50-best translation candidates can be reranked. According to these reranking results, the performance of machine translation system can be evaluated, which also means, the language models can be evaluated indirectly.

We use following two measures for evaluating reranking results.

BLEU (Papineni et al., 2002): BLEU score measures how many words overlap in a given candidate translation when compared to a reference translation, which provides some insight into how good the fluency of the output from an engine will be.

TER (Snover et al., 2006): TER score measures the number of edits required to change a system output into one of the references, which gives an indication as to how much post-editing will be required on the translated output of an engine.

As shown in Table 2, since the results performed by our implementation (3-gram+MKN) is almost the same as that performed by existing language model toolkits IRSTLM⁸ and SRILM⁹, we believe that our implementation is correct. Based on the results, considering both BLEU and TER score, DHWS-3-gram model using GLM smoothing outperforms other models.

5 Conclusion

We proposed an improved hierarchical word sequence language model using directional information. With this information, HWS models can be build more syntactically appropriate while remaining its original advances. Consequently, higher performance can be achieved, both intrinsic and extrinsic experiments confirmed our thoughts.

In this paper, we construct HWS structures (binary trees) based on its original heuristic rule. It is conceivable that more valid discontinuous patterns

Models(+Smoothing)	BLEU	TER
IRSTLM(+MKN)	31.2	49.1
SRILM(+MKN)	31.3	48.9
3-gram(+MKN)	31.3	49.1
3-gram(+GLM)	31.3	49.2
HWS-3-gram(+MKN)	31.2	48.6
HWS-3-gram(+GLM)	31.2	48.7
DHWS-3-gram(+MKN)	31.2	48.6
DHWS-3-gram(+GLM)	31.3	48.6

Table 2: Performance of SMT system using different language models. For the settings of IRSTLM and SRILM, we use default settings except for using modified Kneser-Ney as the smoothing method

can be extracted if we use word association information to built HWS structures, which is a promising future study.

⁷http://www.statmt.org/moses/

⁸http://sourceforge.net/projects/irstlm/

⁹http://www.speech.sri.com/projects/srilm/

References

- B. Allison, D. Guthrie, L. Guthrie, W. Liu, Y. Wilks. 2005. *Quantifying the Likelihood of Unseen Events:* A further look at the data Sparsity problem. Awaiting publication.
- S. Bickel, P. Haider, and T. Scheffer. 2005. Predicting sentences using n-gram language models. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT'05, pages 193-200, Stroudsburg, PA, USA. Association for Computational Linguistics.
- P. F. Brown, J. Cocke, S. A. Pietra, V. J. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. Computational linguistics, 16(2):79-85.
- C. Chelba. 1997. A Structured Language Model. Proceedings of ACL-EACL, Madrid, Spain, 1997, 498-500.
- S. F. Chen and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. Computer Speech & Language, 1999, 13(4): 359-393.
- D. Guthrie, B. Allison, W. Liu, L. Guthrie. 2006. A Closer Look at Skip-gram Modeling. Proceedings of the 5th international Conference on Language Resources and Evaluation, 2006: 1-4.
- S. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. Acoustics, Speech and Signal Processing, IEEE Transactions on, 1987, 35(3): 400-401.
- R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. IEEE, 1995, 1: 181-184.
- C. D. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.
- E. Mays, F. J. Damerau, and R. L. Mercer. 1991. Context based spelling correction. Information Processing & Management, 27(5):517-522.
- K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311-318.
- R. Pickhardt, T. Gottron, M. Körner, S. Staab. 2014. A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser-Ney Smoothing. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, 1145-1154.
- L. Rabiner and B.H. Juang. 1993. Fundamentals of Speech Recognition. Prentice Hall.

- C. E. Shannon. 1948. A Mathematical Theory of Communication. The Bell System Technical Journal, 27: 379-423.
- M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. Proceedings of association for machine translation in the Americas, 2006: 223-231.
- X. Wu and Y. Matsumoto. 2014. A Hierarchical Word Sequence Language Model. Proceedings of The 28th Pacific Asia Conference on Language, Information and Computation (PACLIC), 2014, 489-494.