

Using Dependency Grammar Features in Whole Sentence Maximum Entropy Language Model for Speech Recognition

Teemu RUOKOLAINEN^a, Tanel ALUMÄE^b, Marcus DOBRINKAT^a

^a *Department of Information and Computer Science
School of Science and Technology, Aalto University*

^b *Laboratory of Phonetics and Speech Technology
Institute of Cybernetics at Tallinn University of Technology*

Abstract. In automatic speech recognition, the standard choice for a language model is the well-known n -gram model. The n -grams are used to predict the probability of a word given its $n-1$ preceding words. However, the n -gram model is not able to explicitly learn grammatical relations of the sentence. In the present work, in order to augment the n -gram model with grammatical features, we apply the Whole Sentence Maximum Entropy framework. The grammatical features are head-modifier relations between pairs of words, together with the labels of the relationships, obtained with the dependency grammar. We evaluate the model in a large vocabulary speech recognition task with Wall Street Journal speech corpus. The results show a substantial improvement in both test set perplexity and word error rate.

Keywords. language modeling, dependency grammar, speech recognition, whole sentence maximum entropy model

Introduction

In automatic speech recognition (ASR), the standard choice for a language model (LM) is the n -gram model. The n -gram model is used to predict the probability of a word given its context of $n - 1$ preceding words

$$p(w_i | w_1^{i-1}) \approx p(w_i | w_{i-n+1}^{i-1}). \quad (1)$$

The sentence probability is then obtained as the product of these conditional probabilities

$$p_0(s) = \prod_{i=1}^N p(w_i | w_{i-n+1}^{i-1}) \quad (2)$$

where N is the number of words in the sentence s .

Due to the intrinsic sparseness of the data, n in equation 1 is usually set to 3 or 4 (for English). Therefore, the modeling is based on local dependencies of the language only; the grammatical regularities learned by the model will be captured implicitly within these

short word windows. Consequently, we are interested in explicit modeling of grammatical knowledge.

Existing means of explicit grammatical modeling include structured LMs and syntactic triggers; for an overview by Bellegarda, see [1]. Essentially, these models preserve the conditional framework of equation 1; words are assigned syntactical classes based on some suitable parsing, and the current word is then conditioned on both its syntactical and plain word contexts. Alternatively, the grammatical information can be seen as an additional information source to be combined with the local dependency information captured by the n -gram model. For this approach, a natural framework is provided by maximum entropy (ME) modeling. In our experiments, we apply the Whole Sentence Maximum Entropy language model (WSME LM) introduced in [2,3]. The WSME framework enables us to merge arbitrary computational features into the n -gram model. Effectively, the sentence probabilities given by the n -gram model are scaled according to the features present in the sentence.

In the present work, the grammatical information is captured by features extracted using dependency grammar; dependency grammars are discussed by Tapanainen in [4] and Nivre in [5]. Previously, grammatical features extracted with probabilistic context-free grammars (PCFG) have been used successfully in combination with the WSME models in [6]. Using grammatical features together with N -gram features in a WSME LM, a 22% reduction in perplexity over a baseline n -gram model was reported. However, a relatively small training set of 11K sentences was used and no speech recognition experiments were performed. In this work, we show that a similar increase in modeling accuracy can be acquired using dependency grammars. We use a much larger training corpus than previous work and also report a reduction in speech recognition error rate.

1. Whole Sentence Maximum Entropy language model

The Whole Sentence Maximum Entropy language model (WSME LM) [2,3] is a special case of the general exponential LM. The probability given to sentence s by the general exponential LM is of the form

$$p(s) = \frac{1}{Z} \times p_0(s) \times \exp\left(\sum_j \lambda_j f_j(s)\right) \quad (3)$$

where $p_0(s)$ is the probability of the background n -gram model given by equation 2, Z the normalization term to make the probabilities sum up to 1, f_j the features and λ_j the weights associated with the features. The WSME LM is derived from the general exponential LM by setting the features to satisfy some linear constraints. In particular, we define

$$E_p[f_j] = F_j \quad (4)$$

that is, the expectations of features f_j over $p(s)$ are constrained to specific values F_j . The acquired model $p(s)$ is the model which is most similar to the background model $p_0(s)$ while satisfying the constraints. The similarity of the models is defined as the Kullback-Leibler divergence between their probability distributions. For uniform distribution $p_0(s)$, the obtained $p(s)$ is the maximum entropy solution.

In the case of a binary feature f_j , F_j corresponds to the number of occurrences of the feature in the training corpus. By approximating the sentence probabilities $p(s)$ with the empirical distribution of the training corpus $\{s_1, s_2, \dots, s_M\}$, we obtain

$$E_p[f_j] = \sum_m p(s_m) \times f_j(s_m) \approx \frac{1}{M} \sum_{m=1}^M f_j(s_m). \quad (5)$$

Solving $p(s)$ is equivalent to determining values of the feature weights λ_j ; this can be accomplished using iterative techniques such as iterative scaling, gradient ascent, conjugate gradient or variable metric methods. An empirical comparison conducted on these methods by Malouf in [7] strongly suggested that variable metric methods outperform other methods with respect to model performance as well as computational time. In our experiments, we use the variable metric method style limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm.

A significant computational aspect in training of WSME LM emerges from the evaluation of the expectations in equation 5. This requires a large number of sentence samples from the exponential model $p(s)$. The sampling can be done efficiently using Markov Chain Monte Carlo sampling methods as in [3].

Since WSME LMs require information from the whole sentence to calculate its likelihood, they cannot be used in the first pass of speech recognition. Instead, they are usually applied in a later pass to rescore N -best hypotheses from an earlier pass that uses n -gram LM.

2. Dependency grammar features

We apply the WSME framework with features obtained using the dependency grammar [4,5]. Given a sentence $s = (w_1, w_2, \dots, w_N)$, the dependency parsing results in head-modifier relations between pairs of words, together with the labels of the relationships. The labels describe the type of the relation, e.g. subject, object, negate. These asymmetric bilinear relations define a complete dependency structure for the sentence. An example of a parsed English sentence is shown in Figure 1. Here, we have followed the graphical notation convention used by Nivre in [5]; the dependency arrow points from the head to the modifier.

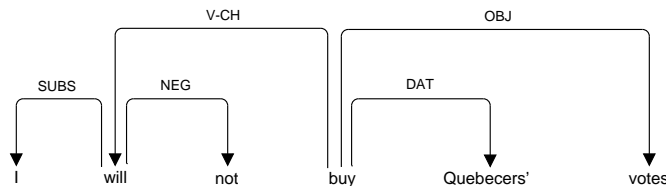


Figure 1. An example English sentence parsed with dependency grammar.

We converted the dependencies into binary features to be used in the WSME model. We experimented with dependency bigram and trigram features. Dependency bigram features contain a relationship between a head and a modifier, together with the name of the syntactic function. Similarly, dependency trigram features contain a modifier with its

head and the head's head, comparable to a child-parent-grandparent relation. Examples of bigram and trigram features extracted from the sentence in Figure 1 are shown in Figure 2.



Figure 2. Example of dependency bigram (left) and trigram (right) features.

3. Experiments

3.1. Data

The textual training corpus consists of news articles from the English Gigaword Corpus¹. The Gigaword is an archive of newswire text data that has been acquired by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. The corpus consists of articles provided by six distinct international new agencies on topics associated with the daily news supply, such as finances, politics and sports. The training corpus was established using 1M sentences (20M words in total) published in articles by Associated Press Worldstream (APW), English Service during 1994-1997, 2001-2002 and 2004-2006.

For speech recognition experiments, we use as a test corpus derived from Wall Street Journal (WSJ) database [8] containing dictated sentences from the WSJ articles. WSJ articles cover financial topics. The test set consists of 329 sentences (11k words in total), spoken by 8 individual speakers.

3.2. Training language models

The training text data is preprocessed as follows. From each article, only the textual body is extracted for further processing (no headlines are included). From the textual bodies, 1M sentences (20M words) are extracted with Perl script². The numerals are expanded into text and abbreviations into complete words. The cases of letters are retained. The case of the first letter of the sentence is decided using majority vote of the remainder of the training data; if the word is seen written with low case elsewhere in the corpus, it is written with low case also in the beginning of the sentence.

Most frequent 60k words of the 1M sentence corpus are selected as the vocabulary for the LM. The pronunciation lexicon for the vocabulary is formed using Sequitur G2P³, a data-driven grapheme-to-phoneme converter [9]. The converter is trained using 50k common words and their pronunciations extracted from the CMU lexicon⁴. For unseen test set of 1k words, the success rate for grapheme to phoneme conversion was 96%.

¹<http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05>

²<http://www.eng.ritsumeikai.ac.jp/asao/resources/sentseg/>

³<http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html>

⁴<http://www.speech.cs.cmu.edu/cgi-bin/cmudict?in=c+m+u+dictionary>

The background LM is a trigram model trained from 1M sentences using modified Kneser-Ney smoothing [10]. Specifically, we train two versions of a trigram model: a regular word-based model, and a model for comma insertion. The latter is a traditional n -gram model that models inter-word commas as regular words. Using the word trigram model, we generate a sample of 5M sentences for training of the WSME model. We then use the comma model to add commas to the sampled sentences by considering the comma insertion a hidden event tagging task [11]. Similarly, during rescoring, we insert commas to the N -best hypotheses provided by the decoder.

To train the ME language model, we first parse the corpus and the sample (with inserted commas) using the functional dependency grammar parser by Connexor (Connexor Machine Syntax⁵). Next, we extract the dependency bigram and trigram features from the parse trees, as described in section 2. The grammatical features are then pruned to only include those that occur at least five times in the generated sample. Excluding features that do not occur in the sample frequently enough is needed in order to avoid infinite gradients for such feature weights during learning, as also observed by Schofield in [12]. In addition to the grammatical features, 18 sentence length features were used, corresponding to lengths $l \in [2..20]$ which activate when sentence length is at least l . The sentence length features were chosen to constrain the marginal distribution of sentence lengths under the model to equal that of the corpus. This is motivated by the fact that an n -gram model lacks explicit modeling of sentence lengths as shown by Schofield.

In the iterative process of estimating the ME model, the normalization term Z in equation 3 was estimated using importance sampling, as in [3]. The L-BGFS algorithm [13], was used for optimizing the parameters. The parameters of the model were smoothed using Gaussian priors. This technique puts a Gaussian (normal) prior centered around zero with a given variance on the model parameters as in [14]. We used a fixed variance value for all parameters and chose it experimentally based on very light tuning on test data.

The ME models were estimated using a set of Python scripts that make heavy use of the Python NumPy and SciPy packages. The training process consumed around 8 GB of RAM.

3.3. Speech Recognition System

The speech recognition system used in the experiments has been developed in the Department of Information and Computer Science at Aalto University (formerly HUT). A thorough description of the system is given in [15]. The main properties of the acoustic modeling are as follows. Speech signal is sampled using 8 kHz sampling frequency and 16 bits. The signal is then represented with 12 MFCC (mel-frequency cepstral coefficients) and the log-energy along with their first and second differentials. Above features are calculated in 16 ms windows with 8 ms overlap. Cepstral mean subtraction (CMS) and a maximum likelihood linear transformation, which is estimated in training, are applied to the features. For acoustic model we use state-clustered Hidden Markov triphone models constructed with a decision-tree method described by Odell in [16]. The model has 5062 states modeled with 32 Gaussians. State durations are modeled with gamma probability functions described in [17].

⁵<http://www.connexor.eu/technology/machinese/demo/syntax/>

Table 1. Perplexity (PPL) and word error rate (WER) when using different language models.

Language model	PPL	WER
Word trigram	303	29.6
WSME LM	244	30.6
Word trigram + WSME LM	255	27.9

3.4. Results

For evaluation, we compared the WSME LM to the background n -gram model using word perplexity and word error rate (WER) results. For the speech recognition experiment, a 1000-best list of hypotheses was generated for each utterance in the test set. We then applied the trained WSME LM to assign new LM scores to all hypotheses, and used the new scores in combination with acoustic model scores and old LM scores for selecting 1-best hypotheses. Perplexity and WER results when using the trigram LM, WSME LM and both models together are given in Table 1.

We observe a 19% relative decline in perplexity (PPL) when using the WSME LM. However, the WER when rescoring the N -best hypotheses using the WSME LM alone is slightly higher than when using the background model. The WER drops by 6.1% relative (1.8% absolute) compared to the baseline when also the background LM scores are used in reranking, although the PPL obtained by the sentence level interpolation of trigram and WSME models is higher than that of the WSME model alone. We suspect that the interpolation with a trigram model provides additional smoothing to the WSME model scores which is probably needed to make the WER results more stable.

4. Conclusions

We described our experiments with WSME LM using binary features extracted with a dependency grammar parser. The dependency features were in the form of labeled asymmetric bilinear relations; experiments were done on dependency bigram and trigram features corresponding to child-parent and child-parent-grandparent relations, respectively. The WSME LM was evaluated in a large vocabulary speech recognition task. We obtained 19% relative improvement in perplexity and 6.1% relative improvement in word error rate when using the WSME LM, compared to a baseline word trigram model.

WSME LMs provide an elegant way to combine statistical models with linguistic information. The main shortcoming of the method is the high memory consumption requirement during training of the model: estimation of a WSME LM using 1M sentences consumed over 8 GB of RAM in our experiment. In practice, tens or even hundreds of millions of sentences are used for estimating LMs for modern speech recognition systems. However, it should be possible to alleviate this problem by profiting from the hierarchical nature of n -gram features, as has been proposed for conditional ME models using N -gram features [18].

Acknowledgments

We thank Ed Schofield who kindly provided the Python scripts that were used for estimating ME models.

This research was partly funded by the Academy of Finland in the project Adaptive Informatics, by the target-financed theme No. 0322709s06 of the Estonian Ministry of Education and Research and by the National Programme for Estonian Language Technology.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under *grant agreement n°* 216529, Personal Information Navigator Adapting Through Viewing, PinView.

References

- [1] J. Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech Communication*, vol. 42, no. 1, pp. 93–108, 2004.
- [2] R. Rosenfeld, "A whole sentence maximum entropy language model," in *Proceedings of the IEEE Workshop on Speech Recognition and Understanding*, 1997.
- [3] R. Rosenfeld, S. Chen, and X. Zhu, "Whole-sentence exponential language models: A vehicle for linguistic-statistical integration," *Computers, Speech and Language*, vol. 15, pp. 55–73, 2001.
- [4] P. Tapanainen and T. Järvinen, "A non-projective dependency parser," in *Proceedings of the fifth conference on Applied natural language processing*, pp. 64–71, Association for Computational Linguistics, 1997.
- [5] J. Nivre, "An efficient algorithm for projective dependency parsing," in *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp. 149–160, Citeseer, 2003.
- [6] F. Amaya and J. Benedí, "Improvement of a Whole Sentence Maximum Entropy Language Model Using Grammatical Features," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, p. 17, Association for Computational Linguistics, 2001.
- [7] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *International Conference On Computational Linguistics*, pp. 1–7, Association for Computational Linguistics Morristown, NJ, USA, 2002.
- [8] D. Paul and J. Baker, "The Design of the Wall Street Journal-based CSR Corpus. February, 1992.," *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1992.
- [9] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.
- [10] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, no. 4, pp. 359–394, 1999.
- [11] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech," in *Proceedings of ICSLP*, vol. 2, (Philadelphia, PA, USA), pp. 1005–1008, 1996.
- [12] E. Schofield, *Fitting maximum-entropy models on large sample spaces*. PhD thesis, Department of Computing, Imperial College London, June 2006.
- [13] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, 1989.
- [14] S. F. Chen and R. Rosenfeld, "Efficient sampling and feature selection in whole sentence maximum entropy language models," in *Proceedings of ICASSP*, (Washington, DC, USA), pp. 549–552, IEEE Computer Society, 1999.
- [15] T. Hirsimäki, J. Pykkönen, and M. Kurimo, "Importance of high-order n-gram models in morph-based speech recognition." *Transactions on Audio, Speech and Language Processing* 2009, Accepted for publication, 2009.
- [16] J. Odell, *The use of context in large vocabulary speech recognition*. PhD thesis, PhD thesis, Cambridge University Engineering Department, 1995.
- [17] J. Pykkönen and M. Kurimo, "Duration Modeling Techniques for Continuous Speech Recognition," in *Eighth International Conference on Spoken Language Processing*, ISCA, 2004.
- [18] J. Wu and S. Khudanpur, "Building a topic-dependent maximum entropy model for very large corpora," in *Proceedings of ICASSP*, (Orlando, Florida, USA), 2002.