ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Towards Recurrent Neural Networks Language Models with Linguistic and Contextual Features

Yangyang Shi, Pascal Wiggers, Catholijn, M. Jonker

Interactive Intelligence, Delft University of Technology, Delft, The Netherlands
yangyangshi@ieee.org, P.Wiggers@tudelft.nl, C.M.Jonker@tudelft.nl

Abstract

Recent studies show that recurrent neural network language models (RNNLM) perform better than traditional language models such as smoothed n-grams. For traditional models it is known that the addition of for example part-of-speech information and topical information can improve performance. In this paper we investigate the usefulness of additional features for RNNLM. We look at four types of features: POS tags, lemmas, and the topics and the socio-situational setting of a conversation. In our experiments, almost all RNNLM models that make use of extra information outperform our baseline RNNLM model in terms of both perplexity and word prediction accuracy. Whereas the baseline model has a perplexity of 114.79, the model that uses a combination of POS tags, socio-situational settings and lemmas achieves the lowest perplexity result of 83.59, and the combination of all 4 types of features, using a network with 500 hidden neurons, achieves the highest word prediction accuracy of 23.11%.

Index Terms: socio-situational setting, part of speech, lemma, topic, recurrent neural networks.

1. Introduction

The goal of a language model is to judge the fluency of a sentence. Statistical language models do so by assigning a probability to every word sequence, or equivalently, by modeling the probability of the next word in an utterance given the preceding words. Statistical language models play an important role in several fields such as automatic speech recognition, machine translation, text classification and optical character recognition.

For a long time smoothed n-grams have been the dominating models in the field. However, it has been shown that recurrent neural network language models (RNNLM) can significantly outperform traditional n-gram models [1, 2].

At the same time, other research showed that linguistic features such as part-of-speech (POS) tags and contextual features such as topic information can improve the performance of smoothed n-gram language models and their generalizations such as hidden Markov models language models, maximum entropy models and dynamic Bayesian network language models [3, 4, 5].

Whereas adding additional features to a n-gram model typically requires hand-crafting relations between different variables and complex smoothing schemes, neural networks are by their very nature suited to combine information from multiple sources. In this paper, we investigate whether the RNNLM can benefit from additional features as well. We experimented with four types of features: POS tags, lemmas, topic information and socio-situational setting information.

The paper is organized as follows. We present related work in the next section. In section 3, we describe the recurrent neural network based language models and the features we used. Section 4 presents the experiments and their results. Finally, conclusions are drawn.

2. Related work

In 2003, Bengio *et al.* [6] proposed a neural probabilistic language model, in which they applied feedforward neural networks. The input to this model are the $n-1$ preceding words, each represented as a real-valued vector.

Recurrent neural network language models [1, 2] avoid this explicit modeling of the word history, by incorporating the time dimension inside the model. Theoretically, recurrent neural networks can store relevant information from previous time steps for an arbitrarily long period of time. This makes learning long-term dependencies possible. Both of these methods demonstrated significant improvements over advanced language models such as Kneser-Ney smoothed n-grams [7].

There has been much work on language models that include additional information. Part-of-speech tags are for example treated as classes for words in class-based language models [8]; as factors in factored language models [9] and as hidden nodes in dynamic Bayesian network based language models [3].

Topic information has been widely used in language modeling, especially in mixture models[4]. Knowledge of the domain is also useful. For example, [3, 10] modeled the socio-situational setting of a conversation as a variable in a dynamic Bayesian network language model.

3. The Model

3.1. Model Structure

Following previous work in this field [2] we used a standard recurrent neural network (Fig 1). It has three layers: an input layer x, a hidden layer h and an output layer y. At time t, the input vector $x(t)$ is constituted by the word vector $w(t)$ and the copied or delayed output $h(t-1)$ from the hidden neurons at the previous time step. In addition we add a feature vector $f_p(t)$ for every feature type p, i.e.:

$$x(t) = [w(t)^T f_1(t)^T \dots f_P(t)^T h(t-1)^T]^T, \tag{1}$$

where P is the number of feature sets added. The word vector $w(t)$ and all feature vectors $f_p(t)$ are 1-of-N encodings of the corresponding word or feature. The output of a neuron i in the hidden layer is:

$$h_i(t) = \varphi(\sum_j u_{ij} x_j(t)), \tag{2}$$

where u_{ij} is the weight between neurons i and j and the activation function $\varphi(z)$ is a sigmoid function:

$$\varphi(z) = \frac{1}{1+e^{-z}}. \tag{3}$$

The final output is:

$$y_k(t) = \phi(\sum_i v_{ki} h_i(t)), \tag{4}$$

where v_{ki} is the weight between hidden neuron i and output neuron k. The activation function $\phi(z_m)$ is a softmax function:

$$\phi(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}, \tag{5}$$

where m is the number of neurons in the output layer, each of which represents a word in the vocabulary. The model is trained, using backpropagation-through-time, to minimize cross-entropy with the training data. As a result, the output layer $y(t)$ represents the probability distribution of the next word given the previous word and the contextual features, which in turn can all be derived from the word history.

3.2. Social and Linguistic Features

3.2.1. POS *tags and lemmas*

Part-of-speech tag sequences provide the model with a limited amount of syntactic information, e.g. that determiners are often followed by nouns. Moreover, adding POS tags to a language model can be seen as a form of smoothing. For word sequences for which there is little or no evidence in the training data, the model can fall back on the more general syntactic classes. In the same way, lemmas can be seen as abstract classes of words. We also added those to our features.

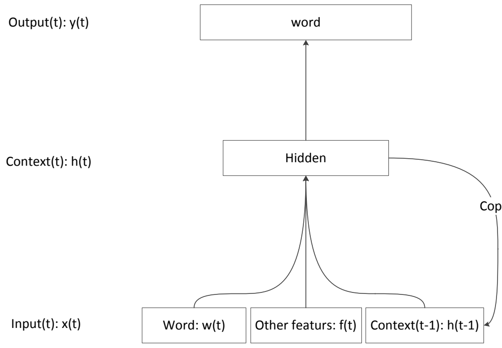

Figure 1: *The recurrent neural network language model.*

3.2.2. *Topic*

The choice of words in a particular conversation is strongly influenced by the topic of that conversation. We automatically derived a set of topics from our set of training files using clustering. First, every document (i.e. the transcription of a conversation) is mapped to a weight vector. This vector contains a weight for every semantically salient lemma type in our vocabulary. The salient lemmas are determined by removing all function words and common content words from the vocabulary. The weights are term frequency-inverse document frequency (TF-IDF) weights, as widely used in information retrieval [11]:

$$\mathrm{w}_{ij} = \begin{cases} (1+\log(\mathrm{tf}_{ij}))\log(\frac{N}{\mathrm{df}_i}) & \mathrm{tf}_{ij} > 0, \\ 0 & \mathrm{tf}_{ij} = 0, \end{cases} \tag{6}$$

where N is the total number of documents and tf_{ij} is the term frequency which counts the number of times lemma i occurs in document j, as high frequency lemmas are thought to be characteristic for the document. df_i is the document frequency which gives the number of different documents in which lemma i occurs, modeling the fact that lemmas that appear in many documents are less discriminative. Both frequencies are logarithmically scaled. A cosine metric is applied to measure the semantic distances between vectors. Clustering was done using k-means clustering.

3.2.3. *Socio-situational setting*

Whereas the topic of a conversation relates to the content of that conversation, the style of the conversation, e.g. whether it is a formal or an informal conversation, relates to the socio-situational setting in which the conversation takes place. The socio-situational setting is characterized by the goal of the conversation, the relationship between speakers and listeners and the number of speakers and listeners. Table 1 shows the 14 different socio-situational settings we use in this paper.

Table 1: Overview of the CGN.

socio-situational setting	words
Spontaneous conversations	2,626,172
Interviews with teachers of Dutch	565,433
Spontaneous telephone dialogues	2,062,004
Simulated business negotiations	136,461
Interviews/ discussions/debates	790,269
Discussions/debates/ meetings	360,328
Lessons recorded in the classroom	405,409
Live commentaries (broadcast)	208,399
Newsreports/reportages (broadcast)	186,072
News (broadcast)	368,153
Commentaries/columns/reviews (broadcast)	145,553
Ceremonious speeches/sermons	18,075
Lectures/seminars	140,901
Read speech	903,043

4. Experiments

4.1. Data

The Corpus Spoken Dutch (Corpus Gesproken Nederlands; CGN) [12] shown in Table 1, is used in our experiments. It is an 8 million word corpus of contemporary Dutch spoken in Flanders and Netherlands, formed by 14 components, each related to a socio-situational setting. In total, it consist of 327 different POS tags and 33271 lemmas. In our experiments, 80% of the data in every component was randomly selected for training, 10% for validation testing and the rest for evaluation. We defined a vocabulary of 44368 unique words. All words in the data that are not in the vocabulary were replaced by an out-of-vocabulary token.

4.2. Results

Table 2 shows the perplexity and word prediction accuracy of the recurrent neural network based language models with different contextual or linguistic features. Here, word prediction accuracy is the ratio of correctly predicted words to the total number of predictions. Were the predicted word is the word with the highest probability output by a model. As shown in [13], word prediction accuracy is a challenging metric for language models.

4.2.1. Baseline models

As is shown in Table 2, the baseline RNNLM, which does not use any additional features, achieves a significant improvement in terms of perplexity over Kneser-Ney smoothed 5-gram language models which are better than Kneser-Ney smoothed 4-gram and 3-gram language models, and class-based n-gram language models, both trained with SRILM [14]. In RNNLM, we apply the default parameters in RNNLM toolkit [1], except the hidden size. In SRILM, we only manipulated the smooth parameters. This improvement of almost 50% is consistent with results shown in [1] for English. The improvement likely stems from the fact that the RNNLM can model long distance dependencies. This also holds for the additional features. A feature may not only benefit the prediction of the current word, but may also benefit next words.

4.2.2. Adding a single feature

The second set of results in Table 2 shows the contributions of each of the additional features individually. In these experiments we used the POS-tags, lemmas and socio-situational settings from the manual annotation available for our data set. The topic of a test document was found by finding the nearest cluster in the training data.

Each of the features can improve the results of the RNNLM. The socio-situational setting (SS) feature reduces the perplexity of RNNLM by 10%. Adding topics (T10 . . . T100, where the number indicates the number of topics) and lemmas gives similar improvements. Variations in the number of topics make little difference, but the optimum lies around 20 topics, which is the setting that we used on all other experiments involving topics. Adding POS tags gives the largest improvement. The RNNLM with POS tags has a perplexity of 90.56, a relative improvement of 21.11% and a word prediction accuracy of 22.62%. This finding is consistent with results for other types of language models, where the inclusion of POS tags may result in relatively large performance improvements. As mention before, a likely reason is that POS-tags introduce a kind of smoothing.

4.2.3. Combining features

Next, we looked at combinations of features. Although most models outperform the baseline RNNLM, only the model that adds topics and POS-tags outperforms the model with only POS-tags added. It reduces the perplexity of RNNLM by 25.05% and achieves a word prediction accuracy of 22.92%.

Arguing that a word is made up of a lemma and a part-of-speech, we created a model in which the corresponding POS-tag and lemma replace the word in the input. As can be seen in Table 2 in the row "RNNLM+POS +lemma (no word)", the performance is similar to that of an n-gram model. Reconstructing word information from the other features complicates the learning task of the neural network.

Even though the model with topics and POS tags does best when combining two features, this behavior is not reflected when adding a third feature. Adding the socio-situation setting slightly reduces performance, while the model with POS-tags, socio-situational settings and lemmas performs better than the RNNLM with POS-tags and topics. It gets the lowest perplexity among all models tested. Compared to the baseline RNNLM it achieves a perplexity reduction of 27.18%.

For the combination of all features (marked as "complete" in the final group of Table 2) we experimented with different sizes for the hidden layer. "300H" use 300 hidden neurons (as did all models in the other experiments); "500H" uses 500 hidden neurons. The results show that increasing the number of hidden neurons from 300 to 500 does not improve the result by much. However, with 500 hidden neurons, the model achieves the highest word prediction accuracy in our experiments.

Table 2: Perplexity and word prediction accuracy (WPA) results.

Model	Perplexity	WPA
SRILM+KN5	228.82	-
SRILM+class+KN5	227.88	-
RNNLM	114.79	20.60
RNNLM+POS	90.56	22.62
RNNLM+lemma	102.26	21.57
RNNLM+SS	103.99	20.84
RNNLM+T10	102.45	21.58
RNNLM+T20	101.07	21.70
RNNLM+T40	104.48	21.51
RNNLM+T100	106.28	21.38
RNNLM+POS +SS	93.63	22.01
RNNLM+POS +T20	86.43	22.92
RNNLM+SS+T20	94.20	22.08
RNNLM+SS+lemma	93.35	22.03
RNNLM+POS +lemma(no word)	230.75	14.37
RNNLM+POS +lemma	90.49	22.63
RNNLM+POS +SS+T20	87.41	22.60
RNNLM+POS +SS+lemma	83.59	22.93
RNNLM+complete(300H)	85.88	22.85
RNNLM+complete(500H)	84.81	23.11

5. Conclusion

In this paper we showed that recurrent neural network language models can benefit from the inclusion of additional linguistic and para-linguistic features. We considered POS tags, lemmas, topics and the socio-situational setting. In the experiments, we used the spoken Dutch corpus (CGN) to test the RNNLM with different combinations of these four types of information. The results show that among the RNNLM with a single feature added, POS-tags give the largest perplexity reduction of 21.11% and a 2% increase in word prediction accuracy. The best perplexity is obtained by the RNNLM with POS-tags, socio-situational settings and lemmas added, which reduces the perplexity of the baseline RNNLM by 27.18% and improves the word prediction accuracy by 2.31%. A RNNLM with 500 hidden neurons that uses all four types of features achieves the highest prediction accuracy, which improves the baseline model by 2.51% absolute.

It should be noted that in these experiments we used the true part-of-speech tags and topic and socio-situational information that was computed offline to investigate the feasibility of the approach. In the future, we will investigate how online prediction of these features from the word history will influence the model and we will integrate the complete language model with a speech recognizer to assess the impact it has on word error rate.

6. References

[1] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010, pp. 1045–1048.

[2] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 5528 –5531.

[3] Y. Shi, P. Wiggers, and C. M. Jonker, "Language modelling with dynamic bayesian networks using conversation types and part of speech information," in *The 22nd Benelux Conference on Artificial Intelligence (BNAIC)*, 2010.

[4] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the Ieee*, vol. 88, no. 8, pp. 1270–1278, 2000.

[5] P. Wiggers and L. Rothkrantz, "Combining topic information and structure information in a dynamic language model," in *Text, Speech and Dialogue*, 2009.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[7] *Improved backing-off for M-gram language modeling*, vol. 1, 1995.

[8] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[9] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2*, ser. NAACL-Short '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 4–6.

[10] Y. Shi, P. Wiggers, and C. M. Jonker, "Socio-situational setting classification based on language use," in *IEEE workshop on automatic speech recognition and understanding*, 2011.

[11] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, May 1999.

[12] O. Nelleke, G. Wim, E. Frank Van, B. Louis, M. Jean-pierre, M. Michael, and B. Harald, "Experiences from the spoken dutch corpus project," in *Proceedings of the third international conference on language resources and evaluation*, 2002, pp. 340–347.

[13] A. van den Bosch, "Scalable classification-based word prediction and confusible correction," *Traitement Automatique des Langues*, vol. 46, no. 2, pp. 39–63, 2006.

[14] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *Proc. Int. Conf. Spoken Language Processing (ICSLP 2002)*, 2002.