# Adaptive Language Modeling with a Set of Domain Dependent Models

Yangyang Shi, Pascal Wiggers, and Catholijn M. Jonker

Interactive intelligence Group, Delft University of Technology
Mekelweg 4, 2628CD, Netherlands
shiyang1983@gmail.com

**Abstract.** An adaptive language modeling method is proposed in this paper. Instead of using one static model for all situations, it applies a set of specific models to dynamically adapt to the discourse. We present the general structure of the model and the training procedure. In our experiments, we instantiated the method with a set of domain dependent models which are trained according to different socio-situational settings (ALMOSD). We compare it with previous topic dependent and socio-situational setting dependent adaptive language models and with a smoothed $n$-gram model in terms of perplexity and word prediction accuracy. Our experiments show that ALMOSD achieves perplexity reductions up to almost 12% compared with the other models.

## 1 Introduction

A language model judges whether a sequence of words is a fluent sentence in a language or not. Statistical language models do so by modeling probability distributions over word sequences. This paper focuses on language models that adapt to the domain to which they are applied.

Current state of the art statistical language models are smoothed $n$-grams, maximum entropy models, and more recently recurrent neural networks [1]. All of these models rely on a relatively small history of previous words to predict the next word in a sentence and do not take into account the larger context of the conversation at hand. In other words, these models assume that the same word distributions can be used in all situations.

However, when dealing with different tasks or situations, people cognitively adapt their language [2]. For example, consider the syntax and lexicon employed in a formal article, versus a casual conversation. In this paper, we study models that take into account the diversity and variability of language over context.

Taking contextual information into account is beneficial to language models, as is shown by topic dependent and socio-situational setting dependent language models [3,4,5,6,7]. Such models typically interpolate between multiple domain specific language models. This leads to robust models that favor coherent discourses. The price for this robustness is the fact that through interpolation the results of component models that match the current discourse may be weighed down by non-matching components. Therefore, a challenge is to create statistical language models that can dynamically select the domain specific models that best fit the current fragment of discourse. In this way, it can avoid the impact of non-matching component.

P. Sojka et al. (Eds.): TSD 2012, LNCS 7499, pp. 472–479, 2012.

Theoretically, better performance can be achieved if a specific model would be available that matches the statistical regularities in the current discourse [8]. For example, a model trained on texts on economics and marketing will get better results in predicting texts from the Wall Street Journal than a general model.

Clearly, the chance of having a model that exactly matches the current discourse is small, but what if we have available a rich set of models that correspond to different topics and/or different types of discourse, can we then dynamically select one model among those models most suited for the current discourse? And would this outperform an interpolation approach? These are the questions investigated in this paper.

The paper is organized as follows. In the next section, related work is discussed. In section 3, we give the theoretical background and motivation of our ALMOSD model, and discuss the structure of the model as well as the procedures we used for training a specific instance of this adaptive language modeling approach. In Section 4, we compare a smoothed *n*-gram model, a topic dependent and a socio-situational setting dependent language model with ALMOSD in terms of perplexity and word prediction accuracy. Finally, based on the results, conclusions are drawn in Section 5.

## 2 Related Work

Several adaptive language modeling approaches have been proposed in the literature. [9] classifies the adaptation methods into three categories according to the underlying philosophy: model interpolation, constraint specification and meta-information (e.g. semantic knowledge, topic knowledge, syntactic infra-structure) extraction.

In model interpolation, [10] proposed the class based *n*-gram language models. Their adaptation strategy is to assign words with similar meaning and syntactic function into one class.

In constraint specification adaptive language models should satisfy the features extracted from the adaptation data. Exponential models trained using a maximum entropy approach, separately assign different weights for each feature [11].

[6] and [12] proposed a mixture language models adaptation method. A collection of sub-models are trained on separate pre-defined domains. Mixture language models linearly interpolate these sub-models. However, as discussed in [13], in actual usage, the mixture language model doesn't work well, partly because of the complicated smoothing. [7,14] explicitly model the domain as a variable in their language models. In this case, it avoids complicated smoothing, as only one model needs to be trained.

In meta-information extraction, there has been much previous research in applying topic information in language modeling [4,6].

All these adaptation methods finally generate one general language model to capture the diversity of natural language. In ALMOSD, we use different models to represent different domains of natural language.

## 3 The Model

As discussed by [15], psychophysical evidence for the existence of parallel processing channels in human processing of speech has been found, especially in dealing with

**Table 1.** Perplexity results of specific sub-models (SM), a smoothed trigram (ST), a topic dependent (TM) and socio-situational setting dependent (SSM) model

| comp | socio-situational setting | SM | ST | TM | SSM |
|---|---|---|---|---|---|
| a | Spontaneous conversations ('face-to-face') | 220 | 222 | 226 | 221 |
| b | Interviews with teachers of Dutch | 196 | 214 | 213 | 212 |
| c&d | Spontaneous telephone dialogues | 188 | 191 | 192 | 190 |
| e | Simulated business negotiations | 110 | 153 | 154 | 152 |
| f | Interviews/ discussions/debates | 274 | 284 | 283 | 281 |
| g | (political) Discussions/debates/ meetings | 287 | 372 | 366 | 349 |
| h | Lessons recorded in the classroom | 298 | 315 | 314 | 313 |
| i | Live (eg sports) commentaries (broadcast) | 275 | 425 | 402 | 369 |
| j | Newsreports/reportages (broadcast) | 345 | 369 | 367 | 361 |
| k | News (broadcast) | 366 | 573 | 560 | 563 |
| l | Commentaries/columns/reviews (broadcast) | 425 | 440 | 435 | 434 |
| m | Ceremonious speeches/sermons | 398 | 444 | 436 | 434 |
| n | Lectures/seminars | - | - | - | - |
| o | Read speech | 573 | 705 | 682 | 695 |

unexpected words. This evidence inspired us to design an adaptive language modeling which can automatically select one from a set of domain dependent models.

The other source of inspiration is the behavior of language models. It is well known that domain-specific language models perform much better on a given domain than general-purpose models. This is illustrated in Table 1, where perplexity results are shown for data from 14 different domains. Perplexity is a measurement of the performance of language models. A better model returns lower perplexity on the same test data set. The perplexity is calculated according to:

$$PP = 2^{-\frac{1}{t}\log P(w_1 w_2 \ldots w_t)}, \tag{1}$$

where $w_1 w_2 \ldots w_t$ is the data in the test set.

The results in the column marked SM in Table 1 are obtained by domain specific models, trained on similar data from the same domain, whereas all results the column marked ST are obtained by a smoothed trigram trained on data from all domains. The columns marked TM and SSM show the results obtained with two sentence-level mixture models the first of which contains topic-specific submodels found by automatic clustering and the second of which uses the domain-specific models of column SM as components (the details of these models will be explained in the next section).

These results suggest that a model that predicts the next word according to a distribution that fits that of a domain specific model will outperform a static model as well as mixture models. So, if $k$ is the current domain and $P_k$ a corresponding model (e.g. on of the models listed in the third column of Table 1), then for an adaptive model $\hat{P}$ the following should hold:

$$\hat{P}(w_i|w_1 \ldots w_{i-1}) = P_k(w_i|w_1 \ldots w_{i-1}), \tag{2}$$

where $w_i$ is the word at position $i$. Note that this formulation allows the selection of a different $k$ for every word in a discourse.

The question then is how to select the right specific model $P_k$ at every point in time. For this purpose we introduce the function $\Phi()$ that predicts the domain $k$ based on the word history:

$$\hat{P}(w_i|\Phi(w_1 \ldots w_{i-1}) = k) = P_k(w_i|w_1 \ldots w_{i-1}). \tag{3}$$

We chose to use the set of specific models themselves to implement this function, i.e. the model that best matches the history seen sofar, is the one used to predict the next word:

$$\Phi(w_1 \ldots w_{i-1}) = \arg\max_k P_k(w_1 \ldots w_{i-1}). \tag{4}$$

An alternative way to think of this model is as a model that puts all word histories that best match a particular model $k$ in one equivalence class. This implies that in general, there is no need to use the sub-models themselves to select the appropriate distribution for prediction, any suitable function of the word history will do. Also, there is no restriction on the submodels that can be used, one could for example include a general purpose model or mixture models as components as well to deal with those cases in which no appropriate specific model is available.

### 3.1 Models Training

All sub-models used in this paper are interpolated trigrams trained with a two phase procedure. For all models the same vocabulary is used. The process of training of the sub-models is as follows:

**Initial Training.** Initially, a unigram, bigram and trigram model are trained on all training data using MLE. Next these models are interpolated:

$$\hat{p}(w_i|w_{i-2}w_{i-1}) = \lambda_1 p(w_i|w_{i-2}w_{i-1}) + \lambda_2 p(w_i|w_{i-1}) + \lambda_3 p(w_i), \tag{5}$$

where $w_i$ is the $i$-th word in a sentence. The interpolation weights $\lambda_1$, $\lambda_2$ and $\lambda_3$ are estimated using a held-out data set. This model is used as our baseline smoothed trigram model (ST) and as a basis for all other models.

**Sub-model Training.** In the second phase, the complete data set is partitioned into a set of sub-domains (Table 1). Each domain specific model is trained on the corresponding subset. The final component based model is obtained by interpolating this model with the general model of phase 1:

$$P(w_i|w_{i-1}...w_{i-n}) = \theta_C P_C(w_i|w_{i-1}...w_{i-n}) + \theta_S P_S(w_i|w_{i-1}...w_{i-n}), \tag{6}$$

where $\theta_C + \theta_S = 1, \theta_C, \theta_S \geq 0$, $P_C$, $P_S$ represent the probability learned in complete training and subset training, respectively.

In initial training the complete data trained models actually are the average of the distributions of each subset. They avoid overfitting in some degree, but they also ignore the characteristics of each subset. The sub-models which are the interpolation of the complete data and subsets data, highlight the distribution of each subset. At the same time, they are controlled by the complete data distribution to avoid overfitting.

# 4 Experiments

## 4.1 Data

The Corpus Spoken Dutch (Corpus Gesproken Nederlands; CGN) [16,17], an 8 million word corpus of contemporary Dutch spoken in Flanders and Netherlands is used in our experiments. This data set is made up of 15 components, each related to a socio-situational setting. The socio-situational settings used in this paper are shown in Table 1.

In the experiment, 80% of the data in every component was randomly selected for training, 10% for development testing and 10% for evaluation. A vocabulary with 44,368 words was created, which contains all unique words that occur more than once in the training data. All words in the test data that are not in the vocabulary were replaced by an out-of-vocabulary token.

## 4.2 Topic and Socio-situational Setting Dependent Language Models

We compared our model with a baseline smoothed trigram, but also with two mixture models: one based on topic-dependent models (TM) that were found by automatic clustering of the data and one based on the same set of socio-situational models that make up the components of our model (SSM). Both are sentence level mixture models [6]:

$$p(w_{1,N}) = \sum_{T} p(T) \prod_{i=1}^{N} p(w_i | w_1 \dots w_{i-1}, t_i), \tag{7}$$

and

$$p(T) = p(t_1) \prod_{i=2}^{N} p(t_i | t_{i-1}), \tag{8}$$

where $t_i$ represents the topics or socio-situational settings at time $i$.

In these models, the current topic or socio-situational setting is dependent on the previous one; the current word is dependent on a history of two words and the previous topic or socio-situational setting. For the details of combining topic information in dynamic language models see [4,6].

## 4.3 Results

Table 2 shows the performance of the models on the entire test set in terms of perplexity. The interpolated models perform only slightly better than the smoothed trigram. ALMOSD clearly outperforms the three other models with a perplexity reduction of 11.91%.

In addition to perplexity, we also use word prediction accuracy to measure the performance of the language models. Word prediction has many applications in natural language processing, such as augmentative and alternative communication, spelling correction, word and sentence auto completion, etc. Typically word prediction provides one word or a list of words which fit the context best. This function can be realized by statistical language models as a side product. Looking at this from the other side, word prediction accuracy actually provides a measurement of the performance of language models [18].

**Table 2.** Comparison of the models in terms of perplexity (ppl)

| model | perplexity |
|---|---|
| smoothed $n$-gram ST | 277 |
| topic-based mixture model TM | 274 |
| socio-situational setting mixture model SSM | 272 |
| ALMOSD | 244 |

**Table 3.** Comparison of the models in terms of word prediction accuracy (wpa) per component of the data set and for the entire test set

| comp | ALMOSD | ST | TM | SSM |
|---|---|---|---|---|
| a | 16.14 | 15.35 | 15.36 | 15.37 |
| b | 14.90 | 14.83 | 14.83 | 14.87 |
| cd | 18.10 | 17.40 | 17.42 | 17.41 |
| e | 19.07 | 18.02 | 18.02 | 17.97 |
| f | 13.98 | 14.22 | 14.25 | 14.27 |
| g | 14.93 | 14.00 | 14.14 | 14.23 |
| h | 13.42 | 13.41 | 13.40 | 13.40 |
| i | 15.70 | 13.44 | 13.48 | 13.53 |
| j | 13.88 | 12.75 | 12.95 | 13.17 |
| k | 18.26 | 15.54 | 15.51 | 15.62 |
| l | 12.38 | 12.86 | 12.86 | 12.98 |
| n | 12.61 | 12.48 | 12.51 | 12.61 |
| o | 12.64 | 11.85 | 11.95 | 11.98 |
| overall | 16.09 | 15.36 | 15.38 | 15.39 |

Table 3 compares ALMOSD, the smoothed trigram model ST, the topic dependent adaptive language models (TM) and the socio-situational setting dependent adaptive language model (SSM) in terms word prediction accuracy. The models are compared on the entire test sets as well as per component of the CGN listed in Table 1. It can be seen that one the entire data set, the ALMOSD model outperforms the three other models. The model also performs best on most individual components. It does especially well on component k that contains broadcast news (a word prediction accuracy of 18.26% vs a prediction accuracy of 15.62% by the socio-situation setting mixture model).

For these results it should be noted that our models were trained on data from the same set of domains that they were tested on, in case of out-of-domain data, a difference between our model and mixture based models is to be expected. However, note that there is no restriction on the set of models that can be included as components. To handle out-of-domain data, one could include for example a mixture model component.

## 5 Conclusion

Arguing that domain specific language models perform better than general purpose models, we propose an adaptive language modeling method called ALMOSD that

combines a set of domain dependent models with a function that selects the most appropriate domain dependent model for the current situation. In particular, for the task of word prediction the function selects that component model that fits the word history better than the other component models. The prediction of the selected model is chosen as the next word prediction of ALMOSD. At every point in time the model that assigns the highest probability to the word history is chosen.

Our experiments we show that ALMOSD is able to reduce perplexity by 11.91% compared to a smoothed $n$-gram model. It also outperforms the other models tested on a word prediction task.

The architecture of ALMOSD makes it easy to experiment with other functions to select the appropriate component model for the current situation. For example, we will experiment with a limited history horizon, e.g., looking back at most 20 sentences. Other ideas are to experiment with (combinations of) dynamic classification functions of domains.

Furthermore, the architecture of ALMOSD also makes it easy to plug in other models, e.g., richer or models for other specific domains than used in our experiments.

## References

1. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH, pp. 1045–1048 (2010)
2. Foster, P., Skehan, P.: The influence of planning and task type on second language performance. Studies in Second Language Acquisition 18, 299–323 (1996)
3. Wiggers, P.: Modelling Context in Automatic Speech Recognition. Ph.D. thesis, Delft University of Technology (2008)
4. Wiggers, P., Rothkrantz, L.: Combining Topic Information and Structure Information in a Dynamic Language Model. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 218–225. Springer, Heidelberg (2009)
5. Iyer, R., Ostendorf, M.: Modeling long distance dependencies in language: Topic mixtures versus dynamic cache models. IEEE Trans. Speech Audio Process. 7, 236–239 (1999)
6. Iyer, R., Ostendorf, M., Rohlicek, J.R.: Language modeling with sentence-level mixtures. In: HLT 1994: Proceedings of the Workshop on Human Language Technology, pp. 82–87. Association for Computational Linguistics, Morristown (1994)
7. Shi, Y., Wiggers, P., Jonker, C.M.: Language modelling with dynamic bayesian networks using conversation types and part of speech information. In: The 22nd Benelux Conference on Artificial Intelligence, BNAIC (2010)
8. Shi, Y., Wiggers, P., Jonker, C.M.: Combining Topic Specific Language Models. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 99–106. Springer, Heidelberg (2011)
9. Bellegarda, J.: Statistical language model adaptation: review and perspectives. Speech Communication 42, 93–108 (2004)
10. Brown, P.F., Pietra, V.J.D., de Souza, P.V., Lai, J.C., Mercer, R.L.: Class-based $n$-gram models of natural language. Computational Linguistics 18, 467–479 (1992)
11. Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modelling. Computer Speech & Language 10, 187–228 (1996)
12. Seymore, K., Rosenfeld, R.: Using story topics for language model adaptation. In: Kokkinakis, G., Fakotakis, N., Dermatas, E. (eds.) EUROSPEECH. ISCA (1997)

13. Adda, G., Jardino, M., Gauvain, J.L.: Sixth European Conference on Speech Communication and Technology, Eurospeech 1999, budapest, Hungary, September 5-9. ISCA (1999)
14. Wiggers, P., Rothkrantz, L.J.M.: Topic-based language modeling with dynamic bayesian networks. In: Proceedings of the Ninth International Conference on Spoken Language Processing, pp. 1866–1869 (2006)
15. Hermansky, H.: Dealing with Unexpected Words in Automatic Recognition of Speech. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 1–15. Springer, Heidelberg (2011)
16. Hoekstra, H., Moortgat, M., Schuurman, I., van der Wouden, T.: Syntactic annotation for the spoken dutch corpus project (cgn). Computational Linguistics in the Netherlands 2000, 73–87 (2001)
17. Nelleke, O., Wim, G., Frank Van, E., Louis, B., Jean-pierre, M., Michael, M., Harald, B.: Experiences from the spoken dutch corpus project. In: Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 340–347 (2002)
18. van den Bosch, A.: Scalable classification-based word prediction and confusible correction. Traitement Automatique des Langues 46, 39–63 (2006)