K-COMPONENT RECURRENT NEURAL NETWORK LANGUAGE MODELS USING CURRICULUM LEARNING

Yangyang Shi, Martha Larson, Catholijn M. Jonker

Delft University of Technology
Intelligent System Department
Mekelweg 4, 2628CD, Delft, NL

ABSTRACT

Conventional n-gram language models are known for their limited ability to capture long-distance dependencies and their brittleness with respect to within-domain variations. In this paper, we propose a k-component recurrent neural network language model using curriculum learning (CL-KRNNLM) to address within-domain variations. Based on a Dutch-language corpus, we investigate three methods of curriculum learning that exploit dedicated component models for specific sub-domains. Under an oracle situation in which context information is known during testing, we experimentally test three hypotheses. The first is that domain-dedicated models perform better than general models on their specific domains. The second is that curriculum learning can be used to train recurrent neural network language models (RNNLMs) from general patterns to specific patterns. The third is that curriculum learning, used as an implicit weighting method to adjust the relative contributions of general and specific patterns, outperforms conventional linear interpolation. Under the condition that context information is unknown during testing, the CL-KRNNLM also achieves improvement over conventional RNNLM by 13% relative in terms of word prediction accuracy. Finally, the CL-KRNNLM is tested in an additional experiment involving N-best rescoring on a standard data set. Here, the context domains are created by clustering the training data using Latent Dirichlet Allocation and k-means clustering.

***Index Terms*—** Recurrent Neural Networks, Language Models, Curriculum Learning, Latent Dirichlet Allocation, Topics, Socio-situational setting.

1. INTRODUCTION

Conventional n-gram language models are known to suffer in the face of data sparseness. Further, they are limited in ability to model long-distance dependencies and are brittle to within-domain variations [1]. Recent studies have shown that a recurrent neural network language model (RNNLM) can outperform n-gram language models [2]. RNNLMs have superior capabilities to reduce the effect of data sparseness, as well as to model long-distance dependencies.

Here, we focus on addressing the problem of within-domain variations in language modeling. Specifically, we exploit curriculum learning [3] strategies to train RNNLMs. As is argued by [4], it is important to start from simple and small in training recurrent neural networks. This approach is inspired by human learning, which starts from simple patterns and moves to complex patterns. The main contribution of this paper is the use of curriculum learning methods for RNNLMs exploiting component models trained by moving from general patterns to specific patterns. The general patterns and specific patterns used in this paper can be viewed as a special form of simple and complex patterns.

Our proposal to use curriculum learning that moves from general to specific is inspired by the observation that the word-usage patterns and, in particular, n-gram occurrence frequencies, vary among different context sub-domains [5]. The dominant patterns reflected by the whole data are different from the dominant patterns observed in the specific sub-domains. General language models are good at learning the overall pattern of the whole training data. However, the specific patterns in the sub-domains of the training data can be ignored by the general models. For example, in the Dutch-language corpus discussed in more detail later, the most frequent bigram in “Lectures/seminars” sub-domain is not ranked into the top 100 bigrams from the whole data.

In addition, neural networks benefit more from new patterns during training, which means that the data that is fed to the network later in the training process contributes more to the final model. Curriculum learning can be viewed as an implicit way of weighting different parts of the whole data. Basically, the later part of the training data receives more emphasis than the initial part of the training data. The complex patterns that are learned after the simple patterns get more weight. By scheduling the specific sub-domain data later in the training process, we can implicitly let the RNNLM put more weight on the specific patterns.

In this paper, we use a Dutch corpus collected from different contexts of language use (referred to as ‘social-situational

978-1-4799-2756-2/13/$31.00 ©2013 IEEE

settings') to investigate the performance of the CL-KRNNLM when it exploits three different curriculum learning strategies: Starting from Vocabulary (SV), Training Data Sorting (DS) and All-then-Specific Training (AS). The corpus as a whole represents a domain and the contexts represent individual sub-domains, which in this case are known during training. We carry out a comparison of specific models with general models to confirm the importance of modeling sub-domains individually and to demonstrate the potential of curriculum learning. In an additional experiment, we investigate the potential of CL-KRNNLM when the sub-domains are unknown. The results confirm that curriculum learning can be applied in case where the sub-domains are not known during training. However, they also reveal that the performance gains of curriculum learning are dependent on the sub-domain variation in the domain to which the CL-KRNNLM is applied.

The rest of the paper is organized as follows. Section 2 discusses related work on RNNLMs, curriculum learning, and language modeling with mixture models. In Section 3, we discuss three methods of curriculum training in constructing specific component models. Section 4 presents the results of experiments in which sub-domain information is known during training. Section 5 presents an additional experiment in which sub-domain information is unknown. The final section concludes.

2. RELATED WORK

Our work is related to previous work about the following areas: recurrent neural networks language modeling, curriculum learning for recurrent neural networks and mixture models, which are covered in this section in turn.

In [6, 7], a sentence level mixture model is proposed, in which the joint probability of each sentence is a linear interpolation of the sentence probabilities from all k component language models. Each component language model is trained on one cluster of the training data. The performance of mixture models depends on the clustering of the training data. In [7], a two-stages clustering process is used. In this paper, we use Latent Dirichlet Allocation [8] with a k-means method to cluster the training data. Considering that too aggressive partition of the training data may aggravate data spareness for component language model training, cf. [7], the mixture probability is interpolated with an additional general model. In this paper, we propose curriculum learning as an alternate to linear interpolation capable of implicitly adjusting the weights between the general and specific data.

Feed-forward neural network language models were proposed in [9]. Each word in the vocabulary is mapped by a shared parameter matrix to a real vector. Mikolov [2, 10] extended the feed-forward neural network language model to a recurrent neural network language model by incorporating the time dimension into the input layer. As is shown in [11], RNNLMs outperform other advanced language models. Their superior capability is derived from their use of a mapping that projects discrete words into a continuous space. Also, the memory they include can be used to model long-distance dependencies. In this paper, we take advantage of the ability of the RNNLM framework to model long-distance context domain information (i.e., socio-situational setting and topic).

Recently, Mikolov *et al.* [12] proposed a context dependent RNNLM, in which context information is obtained from the preceding text using Latent Dirichlet Allocation (LDA). In this paper, we also use LDA in the process of clustering the training data. However, instead of using one general model, we propose to use k-component models trained by curriculum learning. The k-component recurrent neural network language models were first proposed by our previous paper [13], in which the specific component RNNLMs are constructed using interpolation. In this paper, the component models are constructed using curriculum learning.

Curriculum learning for recurrent neural networks has been investigated by [4] from the perspective of the intersection of cognitive science and machine learning. The results presented in [4], which were based on learning the grammatical structure of a language, suggested that it is important to train neural networks using a curriculum such that the training starts with simple patterns and then gradually proceeds to complex patterns. The curriculum learning strategy was recently revisited by [3] and [14]. Their results show that well-designed curricula can benefit RNNLMs for faster convergence, as well as reduced perplexity. Inspired by this work, we also use the curriculum learning, but for a different purpose. Instead of moving from simple to complex, we take advantage of information concerning context to shape the training of a specific component language model from general data to specific data.

3. CURRICULUM LEARNING FOR K-COMPONENT MODELS

In this section, we discuss three different ways of using curriculum learning in constructing the k-component models included in CL-KRNNLM. At first, we describe the basic structure of RNNLM. Then we give the details about the curriculum learning methods used in this paper.

3.1. Recurrent Neural Network Language Models

The recurrent neural network adopted in our work originated with [2]. It has three layers: an input layer x, a hidden layer h and an output layer y. It is characterized by a loop between the input layer and the hidden layer, which plays the role of a short abstract memory that stores previous information. At each time t, the input vector $x(t)$ is constituted by the current word vector $w(t)$ as well as a copy $h(t-1)$ from the previous hidden neurons. The sigmoid function and softmax function are used as the activation functions in the hidden layer and output layer, respectively. The output layer is generally structured in classes. The weight matrix between the input layer and the hidden layer is estimated by backpropagation-through-time (BPTT)[10], which actually unfolds the loop as a deep neural network. When the maximum entropy extension is applied, there is a weight matrix directly connect the n-gram features to the output layer.

3.2. Different curriculum learning setups

In this paper, the following three different curriculum learning strategies for training the component models has been studied. These methods are constructed so that the impact of the overall domain data (i.e., the general data) increases. With 'Start from Vocabulary' the general data only contribute the vocabulary, and in 'Training Data Sorting' the general data contribute in sequence over the course of training, and 'All-to-Specific' the data contribute in full at the beginning of training.

Start from Vocabulary (SV) We first extract the vocabulary from the whole training data. Each component model in the CL-KRNNLM is constructed with the same vocabulary. Each one is further trained by the data only from its corresponding sub-domain. The training is terminated when the component model cannot achieve additional improvement on the validation data, which is selected from the specific sub-domain.

Training Data Sorting (DS) This setup is driven by training data sorting. Each component model in the CL-KRNNLM is trained by the same data except the different sub-domain order with the corresponding sub-domain data at the end. The validation data is drawn from the corresponding sub-domain.

All-to-Specific Training (AS) Each specific component model in the CL-KRNNLM starts with a specified number of epochs of general training in which the specific component models are trained using the whole data. After the general training, the component models are further trained by training data from the corresponding sub-domain. In the general-training period, we choose validation data that covers all the sub-domains. In the specific-training period, the validation data is selected only from the particular sub-domain.

4. EXPERIMENT WITH KNOWN CONTEXT INFORMATION

The experiments in this section are carried out in a scenario in which the sub-domain labels that reflect the context of use (i.e., socio-situational setting) are known at training time. The first experiment in this section involves an oracle condition under which the correct sub-domain label of each sentence of the test data is known to the system during testing. The second experiment investigates the performance of the system when there are no sub-domain labels available for the test data.

4.1. The socio-situational setting data set

In this section, we investigate curriculum learning based on the socio-situational settings of the training data. As is discussed in [5], the word distribution, syntactic structures all vary according to different socio-situational settings.

The Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) [15] contains audio recordings of Dutch spoken by adults in Netherlands and Flanders. Table 1 gives the overview of the CGN data set. It contains nearly 9 million words divided into 13 components that correspond to different socio-situational settings. Components *comp-a* to *comp-h* contain dialogues or multilogues and the components *comp-i* to *comp-o* contain monologues. From the CGN data, we randomly selected 80% for training, 10% for validation and 10% for testing. The test data OOV rate is 3.8%.

Table 1. Overview of the Spoken Dutch Corpus (CGN)

components	socio-situational setting
comp-a	Spontaneous conversations ('face-to-face')
comp-b	Interviews with teachers of Dutch
comp-c&d	Spontaneous telephone dialogues
comp-e	Simulated business negotiations
comp-f	Interviews/ discussions/debates
comp-g	(political) Discussions/debates/ meetings
comp-h	Lessons recorded in the classroom
comp-i	Live (e.g., sports) commentaries (broadcast)
comp-j	Newsreports/reportages (broadcast)
comp-k	News (broadcast)
comp-l	Commentaries/columns/reviews (broadcast)
comp-m	Ceremonious speeches/sermons
comp-n	Lectures/seminars
comp-o	Read speech

4.2. Comparisons of general models with specific models

To compare the performance of the component model in the CL-KRNNLM using these different curriculum learning strategies, we use perplexity (PPL) and word prediction accuracy (WPA). Perplexity is a commonly used metric for measuring language model performance. It is calculated as the geometric average of the inverse probability of the words on the test data. In addition to perplexity, we use WPA [16] as a practical measure for 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.

The performance of the different curriculum learning strategies is shown in Table 2 and 3 in terms of perplexity and WPA. In these tables, all the models have 300 hidden neurons and use 100 classes. The results on these two tables are obtained based on the oracle situation. In other words, each component model is tested on its corresponding sub-domain.

Table 2. The perplexity (PPL) comparison of conventional general RNNLM (base) and component models in CL-KRNNLM using different setups of curriculum training on the sub-domains of the CGN data set.

comp	base	SV	AS	DS
a	82.6	90.6	80.4	81.0
b	104.2	120.2	89.6	91.9
c&d	73.1	81.2	67.8	69.3
e	80.1	62.2	47.8	47.6
f	157.4	180.1	129.2	133.6
g	283.4	245.2	179.4	180.0
h	141.9	177.0	117.6	120.4
i	341.3	189.8	145.8	146.9
j	222.8	338.8	174.3	176.0
k	553.1	292.9	230.3	221.6
l	293.3	486.8	235.5	236.0
n	289.1	411.8	228.3	228.2
o	480.2	328.4	261.3	269.2

Table 3. The word prediction accuracy comparison of conventional general RNNLM (base), component models using linear interpolation (int) and component models in CL-KRNNLM using different setups of curriculum training on the sub-domains of the CGN data set. "acc" denotes the sentence level context prediction accuracy. The percentage symbol $\%$ is omitted for all the numbers in this table.

comp	base	base-int	SV	AS	DS	acc
a	24.0	24.2	23.5	24.3	24.3	81.2
b	20.2	19.4	19.0	21.0	21.0	98.4
c&d	25.5	25.4	24.5	25.9	25.8	20.3
e	24.5	25.0	23.7	25.9	26.6	100.0
f	18.6	18.3	17.4	19.3	19.1	98.6
g	15.9	16.5	16.0	17.7	17.6	96.3
h	20.9	20.5	18.7	21.7	21.6	97.5
i	16.5	19.9	18.8	19.8	19.7	99.7
j	17.3	16.6	13.4	18.5	18.3	67.0
k	14.5	20.0	19.7	19.8	19.9	100.0
l	15.2	13.7	12.8	17.0	17.2	94.8
n	14.8	13.0	12.4	16.3	16.5	93.2
o	14.2	15.6	15.2	16.4	16.3	100.0

As is shown in these two tables, the curriculum learning SV strategy performs worse than the other two curriculum learning methods over all the components. This suggests that although each specific context is characterized by its own style, it is still based on some general patterns that are insufficiently trained by the SV method.

In AS curriculum learning method, we find that good performance is attained. In our experiment, for most components, the component models with ten epochs general training achieve the best performance. All the component models trained by the AS curriculum learning outperform conventional general RNNLMs under the oracle condition.

The performance of component models using DS curriculum learning achieve similar performance as the component models trained by AS. In each epoch of DS training, the component models in the CL-KRNNLM are actually first trained by the general data and then further trained by the specific data. The difference between AS and DS is that AS makes the transfer from general to specific outside of each epoch, while DS makes it happen inside each epoch.

The condition "base-int" involves component models constructed by a conventional linear interpolation method [7]. Each component language model in the mixture model is a linear interpolation of a model trained on the specific domain with a model trained on the whole data. The interpolation weight is tuned on validation data. Table 3 shows that the AS and DS strategies performs better than the conventional linear interpolation method.

Table 2 and 3 reveal that no matter whether linear interpolation or curriculum learning is used, component models, which emphasize specific sub-domain information, outperform general models on a given sub-domain. Especially, on some specific sub-domains which are dramatically different from the general pattern in the whole data, the component models demonstrated substantial improvement over the general models. For example, in the sub-domain for "News", all the component models get over 50% reduction in terms of PPL and more than 30% of improvement in terms of WPA. This case not only indicates that language models are very sensitive to domain changes, but also shows that it is important to take the context domain information into consideration.

4.3. Component model selection

In practice, the context is unavailable in testing. In this paper, we use the sentence level probability maximization to select one component model for each sentence in testing.

The last column "acc" in Table 3 shows the sentence level context information prediction accuracy. In this paper, we determine the sentence s context information labels $C(s)$ as follows:

$$C(s) = \arg\max_{k} p_k(s, h_s), \tag{1}$$

where h_s is the history of sentence s. $p_k(s, h_s)$ is the joint probability of sentence s with its $history$, which is assigned by the k-th component model in CL-KRNNLM.

As is shown in Table 3, nine out of thirteen components attain more than 95% classification accuracy. For at least these sub-domains, the CL-KRNNLM can use the maximum sentence-level probability can achieve similar performance as it achieves in the oracle situation in which the sub-domain of the test data is known.

Table 4 shows the perplexity and WPA results on the CGN data when context (i.e., sub-domain) information is unknown for the test data. The AS curriculum learning method gives the best results. It achieve 13% relative improvement over the conventional RNNLMs in terms of word prediction accuracy.

Table 4. Word prediction accuracy (WPA) result for the Spoken Dutch Corpus (CGN).

models	WPA (%)
base	20.6
base-int	21.3
SV	20.1
DS	22.1
AS	23.2

5. ADDITIONAL EXPERIMENT WITH UNKNOWN CONTEXT

In this section, we apply the proposed CL-KRNNLM to the Wall Street Journal (WSJ) data set. In previous section, the CL-KRNNLM is shown to be able to outperform both the conventional RNNLM and the sentence level mixture models on the CGN data. However, the CGN covers many sub-domains. Each sub-domain is dramatically different from the others. In addition, the data is collected according to different contexts, which means the sub-domain of the training data is known. However, in practice, many data sets do not have known context. In this section, we will investigate the speech recognition performance of the CL-KRNNLM for the WSJ data set using LDA-based topic clustering for the training data.

5.1. WSJ data

In the WSJ, we use 100-best speech recognition lists from the DARPA WSJ'92 and WSJ'93 data sets, as used by [2, 17]. In the 100-best list set, 333 sentences are used as development data for tuning the combination of language models score and acoustic model score (DEV). The rest, 465 sentences, are used for evaluation (EVAL). The oracle WER for the development data and evaluation data are 6.1% and 9.5%, respectively. The training corpus contains 37M words of running text from the NYT section of English Gigaword. The validation data set contains 186K words. The vocabulary size is 194K.

5.2. Results

In this experiment, all the RNNLMs have 200 hidden neurons and 100 classes. The models are trained by 4 times backpropagation-through-time with a block size of 10. When the Maximum entropy method is used, the size of the direct connection matrix, which connect the previous 3-gram information to output layer, is 1 billion.

Table 5 shows the word error rate performance of the CL-KRNNLM for N-best rescoring. The models using the DS curriculum learning of CL-KRNNLM (i.e., DS and ME-DS in Table 5 achieve $0.1\%-0.2\%$ improvement over the conventional RNNLM (i.e., base and ME-base) in terms of WER. Although this improvement is too small to be significant (according to a paired t-test), we note that the gains are comparable in magnitude to those reported on another type of RNNLM extension, namely context based RNNLM [12]. We have tested the maximum entropy extension of RNNLM model in order to shed light on the contribution of the underlying architecture. As can be seen in Table 5, the DS curriculum learning method can improve the baseline model for these two architectures. This suggests that the small improvement of DS can be reproducible under varying architectures.

Table 5. The percent word error rate (WER) on the WSJ data set comparing Keneser-Ney 5-gram language models (kn-5), the conventional RNNLMs (base) the linear interpolation of sentence level mixture models (base-int), the CL-KRNNLMs using different curriculum training methods(SV, DS and AS) and the maximum entropy extensions of all the models ("ME"). "T" column represent the topic number.

models	T	DEV	EVAL	T	DEV	EVAL
kn-5		12.1	17.3			
base		11.4	15.5			
base-int	5	12.0	15.9	10	11.9	16.5
SV	5	11.5	16.3	10	11.8	16.6
DS	5	11.0	15.6	10	11.1	15.4
AS	5	11.3	15.7	10	11.4	15.6
ME-base		10.3	14.9			
ME-base-int	5	11.2	15.3	10	11.3	15.9
ME-SV	5	11.3	16.1	10	11.4	16.3
ME-DS	5	10.4	14.7	10	10.2	15.2
ME-AS	5	10.1	15.1	10	10.3	14.9

Compared with the experimental results on the CGN data, the CL-KRNNLM only achieves a small improvement over the conventional RNNLM on the WSJ data set. This observation suggests that achieving optimal performance on data for which the context is unknown requires careful optimization of the context representation. The latent topic representation that we built might be disadvantaged by the format of the data—the data set contains no document boundary information. The latent topic models are built on sentences which may contain insufficient contents. We also note that the WSJ is a news corpus, there are no great variation in style in the data. It is possible that even under assignment of optimal context, the performance would not reach that achieved on the CGN data. The additional experiment reveals that the move from a context-known condition to context-unknown condition is non-trivial and requires understanding of how to build context classes and the nature of the sub-domains in the data.

6. CONCLUSIONS

In this paper, we proposed a k-component recurrent neural network language model (CL-KRNNLM) that use curriculum learning methods. Each component model was trained using a curriculum that prioritized the training data from its corresponding context domain. Three different curriculum learning approaches have been proposed in this paper, namely, starting from the same vocabulary (SV), data sorting (DS) and all-to-specific training (AS). We compared these approaches using a Dutch corpus, which is labeled with socio-situational setting information. The results under the oracle condition show that each component model in CL-KRNNLM using DS and AS outperforms conventional RNNLM. Especially on the "News" sub-domain, the component models achieve over 50% reduction in terms of perplexity and more than 30% improvement in terms of word prediction accuracy. When the context information is unavailable during testing, CL-KRNNLM still showed better performance than both conventional RNNLM. In an additional experiment, the CL-KRNNLM is further extended to handle the cases in which context information is unknown. The approach use latent topic information, which is constructed using Latent Dirichlet Allocation and k-means clustering method. The CL-KRNNLM using the DS curriculum learning method achieves a small $0.1\% - 0.2\%$ absolute word error rate reduction over conventional RNNLM. This result suggests that the contribution of curriculum learning in CL-KRNNLM is dependent on the nature of the data to which it is applied.

7. REFERENCES

[1] 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.

[2] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of Interspeech*, 2010, pp. 1045–1048.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proceedings of International Conference on Machine Learning*. 2009, pp. 41–48, ACM.

[4] Jeffrey L. Elman, "Learning and development in neural networks: the importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71 – 99, 1993.

[5] Yangyang Shi, Pascal Wiggers, and Catholijn M. Jonker, "Socio-situational setting classification based on language use," in *IEEE workshop on automatic speech recognition and understanding*, 2011.

[6] Rukmini Iyer, Mari Ostendorf, and J. Robin Rohlicek, "Language modeling with sentence-level mixtures," in *Proceedings of the workshop on Human Language Technology*, 1994, pp. 82–87.

[7] R. Iyer and M. Ostendorf, "Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models," in *International Conference on Spoken Language Processing*, 1996, vol. 1, pp. 236–239.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.

[9] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

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

[11] Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan ernock, "Empirical evaluation and combination of advanced language modeling techniques," in *Proceedings of Interspeech*, 2011, pp. 605–608.

[12] Tomas Mikolov and Geoffrey Zweig, "Context dependent recurrent neural network language model," in *IEEE Workshop on Spoken Language Technology*, 2012, pp. 234–239.

[13] Yangyang Shi, Martha Larson, Pascal Wiggers, and Catholijn M Jonker, "K-component adaptive recurrent neural network language models," in *The proceeding of International conference of Text, Speech and Dialogue*, 2013.

[14] Tomas Mikolov, Anoop Deoras, Daniel Povey, Lukas Burget, and Jan Cernocký, "Strategies for training large scale neural network language models," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 196–201.

[15] Nelleke Oostdijk, Wim Goedertier, Frank Van Eynde, Louis Boves, Jean pierre Martens, Michael Moortgat, and Harald Baayen, "Experiences from the spoken dutch corpus project," in *Araujo (eds), Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002, pp. 340–347.

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

[17] Wen Wang and Mary P. Harper, "The superarv language model: Investigating the effectiveness of tightly integrating multiple knowledge sources," in *Proceedings of Conference of Empirical Methods in Natural Language Processing*, 2002, pp. 238–247.