Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 73 (2015) 64–80

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Integrating meta-information into recurrent neural network language models

Yangyang Shi [a,e,*], Martha Larson [a], Joris Pelemans [b], Catholijn M. Jonker [a], Patrick Wambacq [b], Pascal Wiggers [c], Kris Demuynck [d]

[a] *Intelligent Systems Department, Delft University of Technology, The Netherlands*
[b] *ESAT Speech Group, Catholic University of Leuven, Belgium*
[c] *CREATE-IT Applied Research, Amsterdam University of Applied Sciences, The Netherlands*
[d] *Department of Electronics and Information Systems, University of Gent, Belgium*
[e] *Building 18, Xinghu Street 328, Industrial Park, Suzhou 215123, PR China*

Received 28 February 2014; received in revised form 17 May 2015; accepted 5 June 2015
Available online 24 June 2015

Abstract

Due to their advantages over conventional *n*-gram language models, recurrent neural network language models (RNNLMS) recently have attracted a fair amount of research attention in the speech recognition community. In this paper, we explore one advantage of RNNLMS, namely, the ease with which they allow the integration of additional knowledge sources. We concentrate on features that provide complementary information w.r.t. the lexical identities of the words. We refer to such information as *meta-information*. We single out three cases and investigate their merits by means of *N*-best list re-scoring experiments on a challenging corpus of spoken Dutch (referred to as CGN) as well as on the English Wall Street Journal (WSJ) corpus. First, we look at Parts of Speech (POS) tags and lemmas, two sources of *word-level linguistic information* that are known to make a contribution to the performance of conventional language models. We confirm that RNNLMS can benefit from these sources as well. Second, we investigate socio-situational settings (SSSS) and topics, two sources of *discourse-level information* that are also known to benefit language models. SSSS are present in the CGN data, and can be seen as a proxy for the language register. For the purposes of our investigation, we assume that information on the SSS can be captured at the moment at which speech is recorded. Topics, i.e., treatments of different subjects, are present in the WSJ data. In order to predict POS, lemmas, SSS and topic, a second RNNLM is coupled to the main RNNLM. We refer to this architecture as a recurrent neural network tandem language model (RNNTLM). Our experimental findings show that if high-quality meta-information labels are available, both word-level and discourse-level information improve performance of language models. Third, we investigate sentence length and word length (i.e., token size), two sources of *intrinsic information* that are readily available for exploitation because they are known at the time of re-scoring. Intrinsic information has been largely overlooked by language modeling research. The results of both experiments on CGN data and WSJ data show that integrating sentence length and word length can achieve improvement. RNNLMS allow these features to be incorporated with ease, and obtain improved performance.

Keywords: Recurrent neural networks; Language models; Part of Speech; Social–situational setting; Topic; Sentence length

1. Introduction

Language models capture the extent to which a sequence of words can be considered well formed. Most state-of-the-art language models treat language as a

[*] Corresponding author at: Language Understanding Team, Microsoft, Building 18, Xinghu Street 328, Industrial Park, Suzhou, PR China.
E-mail address: yangshi@microsoft.com (Y. Shi).

http://dx.doi.org/10.1016/j.specom.2015.06.006
0167-6393/© 2015 Elsevier B.V. All rights reserved.

sequence of symbols and only make use of the information related to the lexical identities of spoken words. In this work, we focus particularly on language structure manifestations at the word level and at the discourse level. We refer to language-related information that goes beyond the lexical identities of spoken words as *meta-information*. Examples that are relevant to our investigation include word-level meta-information such as Part of Speech (POS) or lemmas and discourse-level information such as the setting in which the speech is delivered (referred to as the social–situational setting) and topic.

Past efforts (Mirowski et al., 2010; Chelba, 1997; Shi et al., 2013; Bellegarda, 1998; Heidel et al., 2007) in language modeling have demonstrated that incorporating additional language-related information at different levels can improve the performance of language models. Conventional *n*-gram language models (Brown et al., 1992; Niesler et al., 1998; Heeman, 1999), however, offer relatively limited possibilities for incorporating meta-information. In order to predict the next word of a word sequence, a conventional *n*-gram language model relies solely on the $n - 1$ words that precede it. This strategy is simple and robust, but is limited in its ability to capture long distance dependencies between words and does not generalize well from sparse data.

Recently, recurrent neural network language models (RNNLMS) (Mikolov et al., 2010, 2011c) have demonstrated potential to address these shortcomings. The success of RNNLMS can be attributed to two factors. First, RNNLMS map the discrete, word-based vocabulary into a continuous space. This mapping makes it possible to learn generalizations over word sequences that are not completely identical, thus reducing the effect of data sparsity. Second, the recurrent loop in the RNNLM architecture, which feeds the hidden layer back into the input layer at every time step, constitutes a memory that serves to capture long-distance dependencies. In this paper, we focus on a third advantage of RNNLMS that has received relatively little attention in the literature. Incorporating meta-information into *n*-gram language models is cumbersome. Generally, it is necessary to design specialized architectures, to create hand-crafted models, or to train weighting parameters. In contrast, integrating meta-information into RNNLMS just requires adding the extra features to the input layer. Viewing recurrent neural networks as a set of logstic regressions helps to make clear that adding extra information can be accomplished elegantly: no special changes to the architecture of the model must be made in order to accommodate the new information.

In practice, RNNLMS are applied in the last pass of a multi-pass speech recognition system. In our experiments this is implemented as an *N*-best list re-scoring task. We choose to work with two data sets. One is drawn from a large and challenging corpus of spoken Dutch (CGN). This corpus contains, by design, very diverse material. In particular, the data has been captured in different social–situational settings (SSSS), i.e., different settings that affect language register and correlate with different topics. In addition to SSS labels, the corpus also contains reference POS and lemma labels. To further demonstrate the performance of the proposed models, the other corpus we choose is Wall Street Journal (WSJ) corpus, which has been used widely in previous work (Mikolov et al., 2010, 2011a; Wang and Harper, 2002; Xu et al., 2009). The WSJ data is news related, which means that the relevant structure in this data set is related to topic. For the WSJ, we use automatic methods to generate topical labels, as well as POS and lemma information.

Our investigations cover three cases of meta-information. First, we investigate *word-level linguistic information*, represented by Part of Speech (POS), tags, and lemmas. Previous work (Heeman, 1999; Wang and Vergyri, 2006) has established that these sources enhance the performance of conventional language models. We confirm that RNNLMS also benefit from these sources. Second, we look at *discourse-level information*, more specifically SSSS and topics. These sources of meta-information are also known to improve conventional language models (Shi et al., 2013; Gildea and Hofmann, 1999; Wiggers and Rothkrantz, 2006b). Here again, we demonstrate their ability to improve the performance of RNNLMS. Finally, a third case concerns meta-information that can be considered *intrinsic*. In other words, the information is inherent in words and word-sequences and does not need to be inferred. Specifically, we investigate sentence length and token size, two features that are readily available for exploitation, but which have been largely overlooked in language modeling. It is difficult to identify a single factor responsible for the lack of attention to intrinsic features in the literature. Most likely, the oversight is due to the combination of the relatively large overhead required to integrate meta- information into conventional language models, already mentioned above, and the a priori impression that intrinsic information is trivial. When RNNLMS are used, the incorporation of extra information is straightforward and elegant, and our experiments demonstrate that trivial information can be exploited to achieve performance gains.

In our investigation, the information on the SSS was captured at the moment at which speech is recorded. As a contrastive condition, we also investigate a setup where no such labels are available, and hence a logic labeling system must be learned in an unsupervised fashion. For this work, we investigate the integration of SSS and topics. The SSS is available for training but not for testing. Topics in this paper are automatically detected using word usage patterns, which are unavailable in training and testing. Therefore, we first use an unsupervised method to obtain the topics for the training data. The topic information obtained by the unsupervised method is further used to train a meta-information predictor that is used to predict topics for test data.

In general, meta-information to be exploited by language models is not known in advance, but rather must be predicted on the fly. Recurrent Neural Networks have shown good results on natural language processing tasks such as named entity recognition and syntactic analysis (Yao et al., 2013; Mesnil et al., 2013). We therefore opted to infer the required meta-information by training an additional recurrent neural network. The RNN that extracts the meta-information feeds into the RNN that models the word sequences (the RNNLMS), resulting in an architecture that we refer to as a Recurrent Neural Network Tandem Language Model (RNNTLM). The first network takes the entries in the *N*-best list as input and outputs meta-information for each word. The output of the first network in combination with the *N*-best word sequence feeds into the main network, which outputs a prediction of the probability of the next word, given the history. We demonstrate how a first RNNLM which predicts POS, lemma, SSS and topic can be coupled to the main RNNLM. Note that RNNTLM is a convenient architecture, and that there is no specific novelty in the nature of the coupling.

Our experimental findings indicate that both word-level and discourse-level information can improve performance. However, in order to obtain a tangible performance improvement, the meta-information must be accurate. In our challenging task, information obtained via unsupervised training did not attain a high enough accuracy, and hence incorporating this information showed little to no improvement. For this reason, we turn to the *intrinsic information* such as sentence length and token size.

The rest of the paper is organized as follows. Section 2 discusses related work on inferring meta-information and on previous methods that have exploited the integration of meta-information into language models. In Section 3, we describe our approach for incorporating meta-information into RNNLMS, including the RNNTLM architecture. Section 4 describes the experimental setup and presents experimental results on the spoken Dutch data and English Wall Street Journal data set. The final section provides conclusions and an outlook.

2. Related work

In this section, we present work related to two key aspects of our approach. First, we survey various forms of meta-information. Next, we discuss previous work that has integrated meta-information into language models and explain how our work builds on and extends these approaches.

2.1. Meta-information

We use the term *meta-information* to refer to information that goes beyond the identity of the word itself. In this section, we briefly survey the types of meta-information that we focus on in this paper.

2.1.1. Word-level meta-information

The word-level meta-information we consider includes Part-of-Speech (POS) tags, lemmas and token size (i.e., word length). As will be discussed in further detail in the next section, POS information improves language modeling. POS tag sequences provide a limited amount of syntactic information to language models. They, for example, allow the language model to capture regularities such as the fact that adjectives are often followed by nouns.

POS information is not an intrinsic property of a word, and for this reason, if it is to be used in language modeling it must be predicted. Both the task of labeling words with POS tags (Antonio et al., 2001; Cutting et al., 1992) and methods to integrate POS with language models (Mirowski et al., 2010; Chelba, 1997) have received considerable research attention. In this paper, the prediction of POS tags as well as the integration of POS tags into language modeling is achieved by using the RNNTLM that we propose here.

A lemma is the set of all word forms that share the same meaning. The citation form of a word that is used in the dictionary represents a lemma. Lemmas provide the language model with morphological information about each word. The number of lemmas is much larger than the number of POS.

Based on compositional morphological representations, Botha and Blunsom (2014) proposed to integrate morphology into language modeling by factorizing each word vector into its surface morphemes vectors. In Mousa et al. (2013), the mixture of words and morphemes along with their features were used as input to Deep Neural Network language models. In Luong et al. (2013), a context aware word representation was constructed by applying Recursive Neural Networks on a morphological binary tree.

Word length, referred to here as token-size (TS), is the size of the word. Here, we measure token size by counting the number of letters in the written form of the word. Token size reflects information about other properties of words. For example, the average token size of content words is bigger than the average token size of function words. Another important characteristic was pointed out by Zipf (1949), namely that token size reflects the frequency with which a word is used in a language. For these reasons, token size is an interesting quantity to divide words into classes. Surprisingly, token size has not been exploited extensively in previous work on language modeling.

Adding word-level information to a language model can be seen as a form of smoothing, especially in conventional *n*-gram language modeling. For word sequences for which there is little or no evidence in the training data, the model can fall back on information concerning the classes to which words belong.

2.1.2. Sentence length (SL)

Sentence length is defined as the number of words in a sentence. It is a indicator of discourse style and genre. This relationship was established even before the advent of the

Digital Age in the field of authorship attribution (Yule, 1939). Recent work observing the relationship between sentence length and genre includes Sigurd et al. (2004) and, for the spoken Dutch data used in this work, Wiggers and Rothkrantz (2007). In particular, sentence length distribution varies for different conversation styles. For example, for spontaneous speech the average sentence length is below 7. In spontaneous face-to-face conversations almost 25% of the sentences contain only one word, such as ‘yes’ or ‘no’ answers and interjections. In contrast, the mean length of sentences in political discussion/debates/meetings is 15, and in ceremonious speeches/sermons, it is 20. In *n*-gram language models and conventional RNNLM, a sentence ending token is explicitly appended to each sentence as a special word in the vocabulary. It helps to capture the kind of words that are likely to occur at the end of a sentence. However, the exact sentence length of a sentence is usually not modeled. An isolated exception may be Bocchieri et al. (2011), who demonstrated that combining separate language models, each created for sentences of different lengths, improves recognition performance in the domain of voice search. In our paper, we aim to exploit the benefits of sentence length in a more general domain.

Applying sentence length information in language modeling can improve the ability of the language model to capture length information. Conventional HMM-based speech recognition systems use a word insertion penalty to prevent the recognizer from overly favoring long strings of short words. However, such a penalty must be tuned on independent data. Our approach allows the system to take sentence length into account without explicit tuning. The more important advantage of our approach to integrating sentence information is that it models sentence length together with content. For example, due to style or syntax, a correlation between lexical items and sentence length can be expected. Our model makes it possible to take this into account.

2.1.3. Topic and socio-situational settings

Both the topic being spoken about and the situation in which language is used—referred to here as socio-situational setting (sss)—impact word distributions. The topic is related to the subject under discussion by the speaker or speakers. In contrast, the sss is more of a proxy for the language register (style of speech), which is influenced by the goal of the conversation, the relationship between speakers and listeners, and the number of speakers and listeners involved. Certain topics may be more typical of some ssss than others, so in general it is not useful to assume that the two are independent. The main distinction in the context of this paper is that the sss can be captured at the time of recording whereas regularities that are discovered automatically in the data are considered to be topic related. In research where topic is expected to mainly reflect the subject matter under discussion, the topic models almost invariably differentiate between topics based on the distribution of content words only, ignoring function words (Putthividhya et al., 2009). In this work, we are interested in modeling underlying clusters in general, and are agnostic if they are related to style or subject matter. Hence, all words are allowed to contribute to the topic model.

Table 1 shows the 14 different ssss of the CGN data used in this paper. Our previous research (Shi et al., 2013) investigated the dynamic classification of ssss using Dynamic Bayesian Networks. In this paper, a recurrent neural network is used to predict the sss and topic for each sentence of the input data. This information is then fed into the RNNLM for the purpose of word prediction and *N*-best re-scoring.

2.2. Language models integrating information beyond word identity

Previous research has established the usefulness of information that goes beyond the identity of words in improving language models. In this sub-section, we survey some of the

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

Components	Socio-situational setting	Words
a	Spontaneous conversations (‘face-to-face’)	2,626,172
b	Interviews with teachers of Dutch	565,433
c and d	Spontaneous telephone dialogs	2,062,004
e	Simulated business negotiations	136,461
f	Interviews/discussions/debates	790,269
g	(political) Discussions/debates/meetings	360,328
h	Lessons recorded in the classroom	405,409
i	Live (e.g., sports) commentaries (broadcast)	208,399
j	News reports/reportages (broadcast)	186,072
k	News (broadcast)	368,153
l	Commentaries/columns/reviews (broadcast)	145,553
m	Ceremonious speeches/sermons	18,075
n	Lectures/seminars	140,901
o	Read speech	903,043

most successful work exploiting this information and explain its relationship to our work.

Decision-tree-based language models (Bahl et al., 1989) are one of the earlier language modeling methods that integrate meta-information with information about word identity. For example, part-of-speech (POS) information can be integrated into language models by asking questions about the word history such as, "Is the last word a verb?" (Heeman, 1999). In Su (2011), the Random Forest Language models of Xu and Jelinek (2004) are extended with morphological, prosodic, syntactic, and topic information.

Class-based language models (Brown et al., 1992) can be viewed as language models that integrate meta-information. Since the quality of the class-based language models depends on how the vocabulary is grouped into clusters, much previous research has been devoted to understanding the best way to cluster the vocabulary (Brown et al., 1992; Ney et al., 1994; Ueberla, 1995; Pereira et al., 1993; Bellegarda et al., 1996; Niesler and Woodland, 1996; Niesler et al., 1998; Yamamoto, 1999). Language models that group words according to POS tag, allow easy integration of POS information with n-gram language models (Ney et al., 1994; Niesler et al., 1998). In this work, we show that automatic determination of the categories yields improved performance over the original POS categories, presumably because it allows control over the category size and composition.

Structured language models (Chelba, 1997; Chelba and Jelinek, 2000) represent another important technique to exploit information beyond the word level. These language models incorporate information concerning the syntactic structure of a language as well as the grammatical function of words in the form of their POS class.

Some language models have integrated meta-information in an effort to better encode information about long distance dependencies between words. Language models incorporating latent topics (Bellegarda, 1998; Heidel et al., 2007) are key examples. These models use a topical representation of the data created by a method such as latent semantic analysis (Bellegarda, 1998) or latent Dirichlet allocation (Heidel et al., 2007). In this paper, we also employ Latent Dirichlet Allocation to construct a representation of documents that captures generalizations over topic. We then create topics by using k-means clustering.

Dynamic Bayesian Networks (DBNS) (Dean and Kanazawa, 1989; Murphy, 2002) offer a concise method to integrate additional features into a language model. Syntactic information, semantic relationships and social background knowledge can be simply specified as a variable into the network structure of the belief network (Wiggers and Rothkrantz, 2006a; Shi et al., 2010, 2011). However, DBNS are generalizations of n-gram language models, and as such share some of their drawbacks. In particular, because they model exact sequences, they tend to suffer in the face of sparse data.

Maximum entropy language models (Pietra et al., 1992; Rosenfeld, 1996) are among the best existing methods for integrating additional information into a language model. These models exploit the maximum entropy principle (Jaynes, 1957) in order to incorporate additional knowledge sources, which can be completely arbitrary in nature. In Rosenfeld (1996) maximum entropy language models using trigger and n-gram features are shown to achieve significant improvement over n-gram language models in terms of perplexity and word error rate. Maximum entropy language models can be viewed as a variety of neural network language models which include no hidden layer. In this paper, we use the maximum entropy extension of the RNNLM (RNNME) proposed by Mikolov et al. (2011c), to incorporate meta-information into RNNLMS. The so called RNNME includes a direct connection between the input layer and the output layer effectively incorporating a maximum entropy language model into the RNNLM architecture.

Neural network based language models, which include feed-forward neural network language models (Bengio et al., 2003) and recurrent neural network language models (Mikolov et al., 2010), are representative of the current state of the art in language modeling. As previously mentioned, recurrent neural network language models are acknowledged for their ability to generalize and their ability to capture long-distance dependencies. Here, we focus on a third advantage, namely their flexible structure, which allows the integration of arbitrary features. Emami and Jelinek (2005) and Alexandrescu and Kirchhoff (2006) investigate the incorporation of syntactic or morphological information into neural network language models.

Factored language models proposed by Bilmes and Kirchhoff (2003) treat each word as a vector of factors. In the work of Wu et al. (2012), the RNNLM is extended to a factored RNNLM. However, in Wu et al. (2012), only word-level information is used. In this paper, we investigate not only word-level information, but also sentence-level and discourse-level information. Furthermore, we investigate the incorporation of intrinsic information such as word and sentence length, which initially gives the impression of being trivial, but actually has the ability to improve language models. The usefulness of integrating topic information, derived via Latent Dirichlet Allocation, into RNNLMS has been studied by Mikolov and Zweig (2012). Here, we significantly expand on both Mikolov and Zweig (2012) and our own previous work on integrating linguistic and contextual information into RNNLMS (Shi et al., 2012). A full range of different types of meta-information is investigated. Further, we go beyond (Shi et al., 2012) in that we evaluate our models applied not only on the task of word prediction, but also on the task of N-best re-scoring. Lastly, we propose a recurrent neural network tandem language model (RNNTLM) which employs RNNLMS both for inferring meta-information and for predicting the probability of the next word.

3. Recurrent neural network tandem language models

3.1. Recurrent neural network language models

The original RNNLMs proposed by Mikolov et al. (2010), consist of three layers: an input layer x, a hidden layer h and an output layer y. RNNLMs are characterized by a loop that integrates a delayed copy of the previous hidden layer into the current input layer at each time step. This loop acts as a short abstract memory that stores previous information. In the hidden layer, the output of a neuron i is:

$$h_i(t) = \varphi\left(\sum_j u_{ij} x_j(t)\right), \tag{1}$$

where the activation function $\varphi(z)$ is a sigmoid function:

$$\varphi(z) = \frac{1}{1 + e^{-z}}. \tag{2}$$

The activation function $\phi(z_m)$ in the output layer is a softmax function:

$$\phi(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}, \tag{3}$$

where z_m is the input of the output layer. with the index m corresponding to one of the words in the vocabulary. The weight $u_{i,j}$ between input layer context part and hidden layer is estimated by backpropagation-through-time (BPTT) (Rumelhart et al., 1986). The loop structure in RNNLM is unfolded by BPTT to a deep neural network. Basically, the RNNLM trained by BPTT is expected to remember information in the hidden layer for several steps.

In Mikolov et al. (2011b), a maximum entropy extension of RNNLMS (RNNMES) is proposed. As shown in Fig. 1, an additional weight matrix directly connects the n-gram features to the output layer,

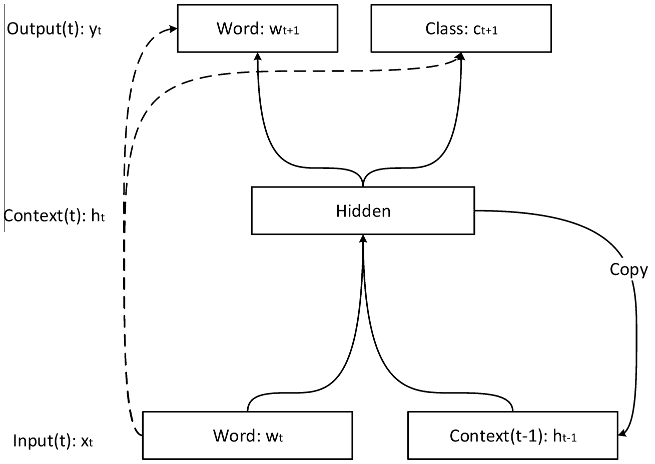

Fig. 1. Class-based Maximum Entropy extension of RNNLMS. The dashed arrows represent the direct connection of n-gram features in the input to the output.

$$p(w|\text{hist}) = \frac{\exp \sum_{i=1}^{N} \lambda_i f_i(\text{hist},w)}{\sum_w \exp \sum_{i=1}^{N} \lambda_i f_i(\text{hist},w)}, \tag{4}$$

where f_j is one feature, λ_i is the weight for feature i and hist is the history of features. The feature f_j includes bigrams (w_{t-1}), trigrams (w_{t-2}, w_{t-1}) up to n-grams. The problem with such a feature representation is that for high order n-grams, it has an impractically large feature set. Most of these features will not show up in the data. So to reduce the complexity of the huge weight matrix connecting the input features to the output layer, a hash function is used to map each n-gram to a single value in a hash array.

$$f(w_{t-2}, w_{t-1}) = ((w_{t-2}) * P_1 * P_2 + w_{t-1} * P_1)\%\text{SIZE}. \tag{5}$$

where P_1 and P_2 are large prime numbers. SIZE is the size of the hash array. % is a modulo function.

In Mikolov et al. (2011c), a class-based RNNLM is proposed. A similar idea has also been investigated in Morin and Bengio (2005). The class-based RNNLM factorizes the output layer using classes. The classes are proportionally determined according to the word frequency in the training data. For example, if we choose M classes, words that take up the top $\frac{1}{M}$ of the unigram distribution would be assigned to class 1. Using a class-based RNNLM, the probability of a word w_{t+1} at time $t+1$ given its history $hist_{t+1}$ is calculated in the following way:

$$p(w_{t+1}|\text{hist}_t) = p(w_{t+1}|c_{t+1}, \text{hist}_t) p(c_{t+1}|\text{hist}_t), \tag{6}$$

where c_{t+1} is the class to which word w_{t+1} belongs. Switching to a class-based RNNLM substantially reduces the computations for updating the weight matrix between the hidden layer and the output layer. Instead of updating a $H \times V$ weight matrix (H is the hidden layer size, V is the vocabulary size), the class-based RNNLM only updates a $H \times C$ weight matrix (C is the class size) connecting the hidden layer with the class part of the output layer as well as a $H \times V_C$ sub-matrix (V_C is the number of words that belongs to class c_{t+1}). As shown in Mikolov et al. (2011c), the class-based RNNLM achieves a 15 times speedup at a cost of 1% accuracy degradation.

3.2. Recurrent neural network tandem language models

Our approach to integrate meta-information into RNNLMS consists of models containing two parts, one part uses a recurrent neural network for predicting the meta-information, and the other part integrates the predicted meta-information into RNNLMS. To predict multiple types of meta-information, several individual recurrent neural networks are used. Recent work (Shi et al., 2015; Collobert et al., 2011) has used multi-task learning to use one model to predict different types of information, but exploration of a single model to predict different types of meta-information lies beyond the scope of the current paper.

Different types of meta-information predictors are needed to extract the various types of meta-information used in this study. Meta-information such as token size and sentence length, is ‘intrinsic’, meaning that it can be derived directly by inspecting the data. Both the token size and sentence length are encoded using the 1-of-N method. In the testing, the unseen token size or sequence length information is ignored by triggering a zero vector. All the words in the same sentence bear the same sentence length information. Sentence length correlates with the topic or the social situational settings of current sentence. Using such an encoding method, we actually cluster the sentences according to sentence length.

However, to obtain word-level information (POS, lemma) and discourse-level information (socio-situational settings and topics) for the test data, we need the aid of a meta-information predictor. In the following subsections, we discuss the two cases in turn.

3.2.1. Integrating word-level meta-information

Word-level meta-information is predicted using the history of the current word. In order to incorporate word-level meta-information, we use the recurrent neural network tandem language model (RNNTLM) architecture that is illustrated in Fig. 2. The meta-information prediction component is an RNNLM as well. In order to predict meta-information m_t for the current word w_t, the previous meta-information m_{t-1}, the current word w_t and the copied hidden layer hm_{t-1} are fed into the network.

$$x_t = [w_t^{\mathrm{T}} m_{1,t-1}^{\mathrm{T}} \dots m_{p,t-1}^{\mathrm{T}} h_{t-1}^{\mathrm{T}}]^{\mathrm{T}}, \tag{7}$$

where p is the number of types of meta-information. The word vector w_t and all meta-information vectors $m_{1\dots p,t-1}$ are represented using a 1-of-N encoding. Because the maximum entropy extension is used, as is shown in Eq. (4), the previous $n-1$ words, the current word, the previous $n-1$ meta-information vectors and the current meta-information vector are directly connected to the meta-information output layer. The sequences of words and the sequence of meta-information vectors are encoded as large hash based vectors using encoding 1-of-N. In this paper, both word sequence and meta-information sequence are represented by hash vectors with 1 billion elements. Note that the input to the maximum entropy extension part is different from the input to the RNNLM part. The input to the maximum entropy part is much larger than the input to the RNNLM. For convenience in Fig. 2, we indicate the maximum entropy extension by using dashed lines that directly connect the input layer of the RNNLMs to the output layer of the RNNLM. The maximum entropy extension integrates additional regular n-gram features into the RNNLM. The fixed and limited length of the n-grams turns out to be complementary to the variable-length history generalizations learned by the recurrent connection of the RNN, allowing the RNNME to capture local regularities. In the output layer of the meta-information predictor, the meta-information $m^*{}_t$ that obtains the highest probability is selected and encoded in a 1-of-N representation, which is fed to the RNNLM:

$$m^*{}_t = \arg \max_{m_t} p(m_t | w_t, m_{t-1}, \mathrm{hist}_{t-1}). \tag{8}$$

When the meta-information is unknown, the current predicted meta-information $m^*(t)$ is copied to the input of the meta-information predictor in order to predict next meta-information $m^*(t+1)$.

As shown in Fig. 2, the part above the horizontal dashed line is also a recurrent neural network. It uses the current word w_t, and the predicted meta-information for the current word $m^*{}_t$ and the copied hidden layer hw_{t-1}, to predict the next word w_{t+1}. In the proposed RNNTLM, the structures of the two recurrent neural networks are basically the same; they differ only in their input and output.

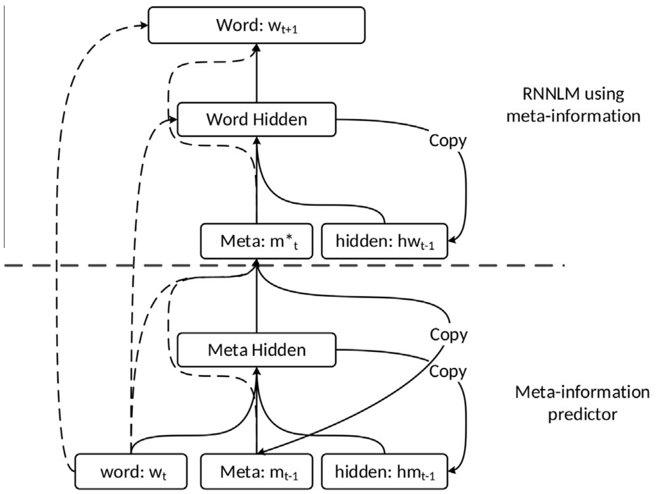

Fig. 2. Recurrent neural network tandem language models integrating word-level information. The RNN below the dashed line predicts meta-information on the word level. The RNN above the dashed line incorporates the meta-information. The dashed arrows represent the direct connection of n-gram features in the input to the output.

3.2.2. Integrating discourse-level meta-information

Discourse-level information is predicted using a recurrent neural network as well, when the information is not available in testing. Because the RNNLM is applied within an N-best rescoring framework, segment information is available at the moment the language model is applied. This information has been generated by the speech recognition system that produced the N-best lists. By looking at the full segment instead of only at the preceding words (cf. Fig. 1), a better prediction of the discourse-level information can be obtained.

We test under known and unknown conditions. Under the ‘known’ condition, the information about the correct discourse-level meta-information category is available at test time. Under the ‘unknown’ condition, the information must be predicted. In order to predict discourse-level meta-information, we train one sub-domain-specific RNNLM for each SSS or topic (also referred to as a component). This model is trained using the curriculum learning method for training domain-adapted RNNLMS. Our previous work has demonstrated the effectiveness of this method for creating RNNLMS for a heterogeneous domain that is composed of a number of sub-domains (Shi et al., 2014). Curriculum learning (Bengio et al., 2009; Elman, 1993) makes use of the fact that neural networks are sensitive to the order in which data is presented to them during training. By presenting the RNNLM first with general domain data and only later in the training phase with sub-domain data, we create models which emphasize the patterns in the sub-domain data. Curriculum learning can be regarded as a form of implicit interpolation between a domain model and a sub-domain model. It achieves the same goal as conventional linear interpolation, but does so with a single, continuously trained model that dispenses with the need to explicitly train weights of individual sub-models. Discourse-level labels, which are predicted at the segment level, are duplicated for each word.

The first type of discourse-level information that we consider is SSS. As previously mentioned, this information is captured at the time the speech is recorded and hence is available to train the language model.

The predicted discourse level meta-information $c(s)$ of a sentence s is derived from the probabilities returned by the different component models as follows:

$$\begin{aligned} c(s) &= \arg\ \max_k p(k|s) = \arg\ \max_k \frac{p_k(s)p(k)}{p(s)} \\ &= \arg\ \max_k p_k(s)p(k), \end{aligned} \tag{9}$$

where $p(k)$ is the prior distribution of different discourse level meta-information assumed to be uniform distribution in this paper. We assume a uniform distribution, since we are interested in avoiding the assumption that the distribution of classes in the target data matches that of the training data. $p_k(s)$ is the probability of segment s given by the kth component model. Each component model is an RNNLM, so the probability of segment s is calculated as follows:

$$p_k(s) = p_k(w_0)p_k(w_1|w_0)\dots p_k(w_t|w_0,\dots,w_{t-1}), \tag{10}$$

where $p_k(w_t|w_0,\dots,w_{t-1})$ is the output of the kth component RNNLM for word w_t.

For each word, the predicted discourse-level meta-information for the segment to which that word belongs is fed into the language modeling part of the RNNTLM and used to predict the next word.

The second type of discourse-level information considered in this work are topics. The topics are derived automatically from the training data using Latent Dirichlet Allocation (LDA) (Blei et al., 2003) in conjunction with k-means clustering. LDA is a probabilistic model that describes the generation process of documents. A document is considered to be a mixture of underlying topics that give rise to the words it contains. LDA applies a bag-of-words strategy, allowing each document to be represented as a latent topic vector whose components reflect the relative contributions of the individual latent topics. We choose to make use of LDA since it represents the state of the art in topic representations. We construct latent topic representations by considering each segment to be a document. We then apply k-means clustering in order to cluster the data. The result is a set of topic clusters. Each word in a segment bears the topic label of the cluster the segment belongs to.

4. Experiments and results

In this section, we describe the setup used to carry out our experiments on the CGN data set (Dutch) and the WSJ (English) data set, and present the results of our experimental investigation. We start out by providing some overarching information concerning the two sets of experiments. The goal of the experiments is to investigate the added value of adding meta-information to RNNLMS, and to gain insight into which meta-information is most useful in which situations. As mentioned above, we are interested in two scenarios meta-information 'known', in which the meta-information is available at test time, and 'unknown', in which the meta-information must be predicted. This comparison allows us to understand the impact of meta-information prediction errors on our RNNLM language models. For intrinsic features, token size and sentence length, are the same for both conditions, since they are trivial to compute. For word-level and discourse-level features, the CGN data set offers the possibility to directly compare 'known' and 'unknown', since the data set includes ground-truth for POS, lemma, and SSS. The WSJ data set does not include similar ground truth. For this reason, we use the Stanford CoreNLP tools, as a high quality method to predict word-level meta-information, POS and lemma. We use ground-truth meta-information directly at test time, to emulate the 'known' condition, and we use it as training data for the 'unknown' condition that uses the RNNTLM approach to predict meta-information. For both the CGN and WSJ data sets, we use topics, automatically discovered by the process described below in order to experiment with discourse-level information. In the case of CGN, these topics provide us a contrast with SSS meta-information, provided with the data set.

4.1. Evaluation metrics

We evaluate our language models in terms of perplexity (PPL), word prediction accuracy (WPA), and word error rate (WER). Both the PPL and WPA are calculated using the language model directly. In other words, the speech recognition system, which is described in Section 4.2.4, is not involved in calculating these evaluation metrics. PPL is the geometric average of the inverse probability of the words on the test data. WPA (van den Bosch, 2006) is a practical measure of language models. It is defined as the accuracy achieved when the language model is provided with information about preceding words and required to predict the word that would occur next. Word prediction is important for natural language processing tasks, such as spelling correction and auto completion. WER is evaluated by carrying out a rescoring experiment that takes as input the *N*-best list generated by the speech recognition system. In this situation, all the meta-information is unknown beforehand.

4.2. CGN experiment

4.2.1. Data

In this section, the language model training and test data comes from the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) (Oostdijk et al., 2002), which contains recordings of standard Dutch spoken by adults in the Netherlands and Flanders in a variety of language usage settings. As shown in Table 1, the entire corpus contains nearly 9 million words divided into 14 components. We used the component as a proxy for the socio-situational setting. Each component is further divided into segments that contain one or more sentences. Segments may be as large as 1000 words.

Components a–h contain dialogues or multilogues and components i–o contain monologues. Our experiments are carried out on a test set that contains 10% of the data randomly selected from components h, g, n and o. The choice of these components was made by practical considerations, which included the need to exclude the data used to train the acoustic models for the speech recognition system that generated the *N*-best list (further described in Section 4.2.4).

In total the test set contains 974 K running words and 149 segments. For language model training, 80% of the CGN data, mutually exclusive from the test set, was used. Another mutually exclusive set of 10% of the data was used for validation. The details of the training data size for each part of the speech recognition system is described in Table 2.

4.2.2. Part-of-Speech and lemma prediction

CGN provides (manually verified) Part of Speech (POS) tags and lemmas for each word (Van Eynde, 2004). There are 281 POS tags represented in the training data. The POS tags consist of a basic set (i.e., including ‘noun’, ‘adjective’, ‘verb’) enriched by further information. Examples of the further information include, for nouns, the type of noun (common noun or proper noun), the number (plural or singular), the degree (whether or not the noun is diminutive), and case (e.g., genitive or dative). The RNNLM trained to predict parts of speech achieves an accuracy of 93.5 when 180 hidden units are used. Changing the number of hidden units has negligible impact on the performance.

Table 2
CGN training data size for different models. “AM” represents training data size for the acoustic model. “*N*-best LM” gives the training data size for the language model that generates *N*-best list. “Rescoring LM” gives the training data size for all the second pass language models.

AM	*N*-best LM	Rescoring LM
CGN comp-c, d, f, i, j, k,l	12 Southern Dutch newspapers 10 Northern Dutch newspapers CGN audio transcription	CGN audio Transcription
115.5 hours audio	1463.7 millions of words	7.2 millions of words

The process of lemmatization involves mapping the inflected forms of words, as they occur in text, to their basic forms, i.e., the way that the word would be cited in the dictionary. Several forms of a word map to the same basic form, for example, the singular and the plural of a word both map to the singular form. There are 84 K lemmas (also pluralized ‘lemmata’) represented in our training data. Note that although many lemmas can be uniquely determined by inspecting the form of a word token, there exist some word tokens in the Dutch language that are ambiguous. In these cases, the context must be considered in order to determine the correct lemma. Because of the large number of lemmas that must be predicted, we use a class-based recurrent neural network (Mikolov et al., 2011c) to speed up the training of the lemma predictor, the classes are determined according to the lemma frequency in the training data. Prediction proceeds in two steps. First, we predict the class using Eq. (11).

$$cl^* = \arg\max_{cl} p(cl|w). \tag{11}$$

Then, we predict the lemma according to the predicted class using Eq. (12).

$$\text{lemma}^* = \arg\max_{\text{lemma}} p(\text{lemma}|w, cl). \tag{12}$$

Using a hidden layer with 30 neurons, the RNN based lemma predictor achieves a 96.3% prediction accuracy. Increasing the number of hidden units causes the performance to decay slightly, attributable to the relatively smaller number of examples available per lemma in the training data.

In the experiments, we test the ‘known’ condition (i.e., oracle condition) in which the POS or the lemma label for a word is known at test time, as well as the ‘unknown’ condition for which the labels are predicted at test time. Under the POS and lemma ‘unknown’ condition, the lower network in RNNTLM uses the word inflected form to predict its corresponding POS and lemma that is integrated with the word inflected form in the upper network for language modeling. In our previous work (Shi et al., 2012), arguing that a word is made up of a lemma and a POS, we created a model in which the corresponding POS-tag and lemma replace the word in the input. Under the oracle situation, such a model did not achieve an improvement over *n*-gram language models, and we do not test it further here.

4.2.3. Socio-situational setting and topic prediction

Both sss and topic we mentioned in this paper are simply ways of dividing the data set into subsets that can be helpful. The sss was collected when the data was recorded (see Section 4.2.1). The topic information is derived by LDA and clustering was described at the end of Section 3.2.2. Note that the notion of topic used in this paper may also contain other information (e.g., style). We refer the output from LDA and clustering algorithm as topic because these algorithms are generally used to find topics. Prediction of sss and topic for the *N*-best lists returned by the recognizer is carried out using the RNNLM-based prediction method described in Fig. 2. It is important to note that this method achieves good performance in predicting sss. The average accuracy on the test data (that covers all of the components in the CGN corpus) is 76.2%. As shown in Table 3, nine out of thirteen components achieve above 95% accuracy. The performance for the components that are used for *N*-best lists is: 98%, 96%, 93% and 100% for components h, g, n and o, respectively. The main challenge for our sss classifier comes from the components c and d, which achieves the lowest accuracy.

4.2.4. Generating the N-best list

For the automatic speech recognition (ASR) experiments we used a Large Vocabulary Continuous Speech Recognition (LVCSR) system. The system, which is an updated version of Demuynck et al. (2009), was built by ESAT using their state-of-the-art ASR toolkit SPRAAK (Demuynck et al., 2008; Demuynck, 2001). It was initially developed for the Dutch *N*-best evaluation benchmark (Kessens and Leeuwen, 2007). The system is a speaker independent speech recognizer that has the capability to select components and adjust parameter settings on the fly, based on observed conditions in the audio.

The acoustic models employ 49 three-state acoustic units (46 phones, silence, garbage and speaker noise) and one single-state phone (short schwa), which are modeled using SPRAAK's default tied Gaussian approach. Under this approach, the density function for each of the 4 K cross-word context-dependent tied states is modeled as a mixture of an arbitrary subset of Gaussians drawn from a global pool of 50 K Gaussians. The mixtures use on average 180 Gaussians to model a 36 dimensional observation vector of MIDA features (Demuynck, 2001). These were obtained by means of a mutual information based discriminant linear transform (MIDA) on vocal-tract length normalized (VTLN) and mean-normalized MEL-scale spectral features and their first and second order time derivatives. The acoustic models are trained on Broadcast News (components f, i, j, k, l in Table 1) and Conversational Telephone Speech (components c and d in Table 1).

Using a lexicon of 400 K words, 5-gram language models (LMs) with modified Kneser–Ney discounting were trained on 4 main text components: 12 Southern Dutch newspapers, 10 Northern Dutch newspapers and transcriptions of broadcast news (component f, i, j, k, l in Table 1) and conversational telephone speech (component c and d in Table 1) (Northern Dutch refers to the Dutch spoken in the Netherlands; Southern Dutch refers to the Dutch spoken in Belgium). This training data set does not overlap with the test data we used in all the experiments. The four LMs were interpolated linearly and perplexity minimization was done to find the optimal interpolation weights on the *N*-best development data. Lexicon creation was handled by an updated version of the system described in (Demuynck et al., 2002). Dutch has a substantial number of (regional) pronunciation variation, which was addressed by using phonological rules to generate the likely pronunciation variants. This resulted in a median of 3.8 pronunciations per word or 1.13 variants per phone in the canonical word transcriptions.

Since Dutch compounds are always written as a single word, the word recognition results are post-processed for compounding. Two subsequent words are replaced by their compound if the following criteria are met: (1) the words are longer than 3 letters, (2) the words are not very rare, and (3) the unigram count of the compound is higher than the bigram count of the individual words. This approach effectively extends the 400 K lexicon to a 6 M lexicon.

The main parameters of the system control hypothesis pruning and combining the language model and the acoustic models. To combine the model scores, we employ our

Table 3
Confusion matrix of socio-situational setting prediction using the meta-information predictor in RNNTLMS on CGN data. Each number in the table is percentage. The percent sign (%) is omitted in the table.

Com	a	b	c and d	e	f	g	h	i	j	k	l	n	o
a	81						19						
b		98			2								
c and d	12		20	1			67						
e				100									
f					99						1		
g						96	4						
h	2						98						
i								100					
j					13		20		67				
k										100			
l										5	95		
n					7							93	
o													100

standard way of handling this problem (Demuynck, 2001), by having a LM scaling factor and a word startup cost. Beam search pruning was applied to control the number of hypotheses in the search space (Steinbiss et al., 1994): a threshold indicates how much the score of a hypothesis can drop below the score of the most likely hypothesis; if most hypotheses have a similar score, a beam width parameter is applied to indicate how many hypotheses can be retained, keeping only the best ones.

Adopting the pruning parameters that yield recognition in real time, we create a lattice with the most likely word sequence hypotheses for each speaker turn in each component. Using SRI's lattice tool, each lattice is converted into an *N*-best list containing the 10 K best sentences, disregarding filler words and silences.

4.2.5. Re-scoring the N-best list with the RNNLMs integrating meta-information

The RNNLM models that we test in our experiments all use the maximum entropy extension (RNNMES), as mentioned in Section 3. They use 300 hidden neurons and one weight matrix with 1 billion elements that directly connect input to output. All the models are trained using Backpropagation Through Time (BPTT) with 5 steps.

4.2.6. Experimental results

In this section, we present the results obtained with our proposed approach for integrating meta-information into RNNLMS. We compare our models to two baselines, KN5-GRAM, which is a conventional Kneser–Ney 5-gram language model, and also RNNME, which is the same RNNLM that we use in our approach, except for the fact that it does not integrate any meta-information. As shown in Table 4, when applied to the task of *N*-best rescoring, the conventional Kneser–Ney 5-gram language model achieves a WER of 40.1 on our CGN data set, and is outperformed by the RNNME language model, which achieves a WER of 38.7. The experiments investigate two scenarios: the 'known' condition, which is an oracle condition under which the information is known at test time, and the 'unknown' condition, under which the RNNTLM architecture in Fig. 2 is used to predict the information at test time. Note that these two conditions are the same for the baselines, which do not integrate any meta-information.

Table 4
RNNLM language models integrating a single feature: perplexity (PPL), word prediction accuracy (WPA) and word error rate (WER) results on CGN data under the condition that meta-information is known and unknown (i.e., predicted) during testing.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
KN5GRAM	140	–	140	–	40.1
RNNME	112	21.3	112	21.3	38.7
POS	97	22.8	104	22.0	38.9
Lemma	109	21.8	114	21.7	38.3
SSS	105	22.2	107	22.1	37.8
T30	96	22.9	118	21.3	38.6
TS	110	21.7	110	21.7	38.2
SL	109	21.9	109	21.9	37.9

First, we discuss our experimental results with models that integrate word-level information, i.e., information on parts of speech and lemmas (cf. the lines labeled 'POS' and 'lemma' in Table 4).

Looking at the WPA for the 'known' and the 'unknown' conditions, we see that both improve over the RNNLM baseline. The WPA gain is less when the meta-information must be predicted at test time (i.e., under the 'Unknown' condition). In the lemma case, the improvement translates into an improvement in WER when rescoring. However, adding POS information slightly damages rather than improves WER performance. In summary, the contribution of word-level information is very modest. However, these results suggest that errors introduced by meta-information prediction do not necessarily have a large impact when compared to the theoretical performance achievable under the oracle condition.

Next, we turn to the experimental results using integrate discourse-level information, i.e., social situational settings and topic (cf. the lines labeled 'sss' and 'T30' in Table 4). For the model integrating information on sss, we see that whether sss labels are known at test time, or must be predicted has relatively little impact on the WPA (cf. 'known' vs. 'unknown'). In both cases, the integration of sss information achieves an improvement in WPA over the baseline RNNME. This improvement also translates into a reduction in WER in the *N*-best rescoring experiment. In contrast with the sss case, limited improvement is achieved in the T30, i.e., the model integrating automatically created topics. Under the oracle conditions, i.e., the topics are known at test time, an improvement in WPA can be achieved. However, the results under the 'unknown' condition are inconclusive.

Note that we know the topic labels for the 'known' condition because the topics were created by clustering all the data simultaneously into topics. This point has two implications. First, the topics of the oracle condition were created on more data (all data in one segment of the CGN database) than the topics of the 'unknown' condition. Second, for the 'unknown' condition, the decision of topic membership was made using only one 'sentence' as returned by the *N*-best list module. Both these factors can explain the gap between the performance under the 'known' and 'unknown' conditions in the case of T30.

An interesting consideration, is the sensitivity of the system to the number of topics. Results of experiments exploring this issue are reported in Table 5. The improvement offered by integrating topics can be seen to vary with the number of topics chosen, reaching its maximum with 30 topics. Taken together, these results suggest that the robustness of topic prediction and the optimization of the number of topics are both aspects that must be taken into

Table 5
RNNLM language models integrating topic information, for different numbers of topics (10–40): results on CGN data under meta-information ‘known’ and ‘unknown’ conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
T10	102	22.5	121	20.7	38.9
T20	99	22.7	126	19.7	39.7
T30	96	22.9	118	21.3	38.6
T40	98	22.7	121	21.0	38.7

consideration when integrating topic as meta-information into an RNNLM.

In summary, the results of the experiments integrating discourse-level meta-information suggest that SSS has potential to improve RNNLMS. If such information has been captured at recording time, it can be used, either directly on the test data, or for training SSS predictors. In the cases in which no information has been captured at recording time, topic discovery can be applied, but it is challenging to exploit it productively.

Now, we turn to the topic of integrating ‘intrinsic’ meta-information into RNNLMS, i.e., token size and sentence length (cf. the lines labeled ‘TS’ and ’SL’ in Table 4). Recall that intrinsic meta-information is particularly interesting since its use has been largely overlooked in the literature on conventional language models. It is ‘free’ information in the sense that it can be derived directly, without the need for prediction. Note that because intrinsic information uses counts of letters in words (TS) and of words in the utterance being rescored (SL), the results of the ‘known’ and the ‘unknown’ condition are the same. Both with respect to WPA and with respect to WER, intrinsic meta-information is able to achieve performance improvement over the RNNME baseline. The performance is slightly better in the case of SL than in the case of TS. It is particularly striking that the WER falls by 0.5 absolute in the case of TS (38.7 to 38.2) and by 0.8 absolute in the case of SL (38.7 to 37.9). In summary, these results suggest that intrinsic information, although trivial to derive, should not be considered trivial when it comes to integrating meta-data into RNNLMS. Instead, this sort of ‘free’ information should be exploited. It is capable of yielding a performance improvement of the same magnitude of the one attainable by more costly methods that require the training of a meta-information predictor.

Finally, we turn to the experiments that integrate multiple features simultaneously into RNNLMS. We first consider experimental results obtained when adding one other feature to a selection of the conditions in Table 4. Results are reported separately for easy comparison in Tables 6–9. We choose POS (Table 6) as a representative word-level feature, because we are interested in whether additional features can close the gap between the ‘known’ and ‘unknown’ condition. We choose SSS (Table 7) as a representative discourse-level feature, because we are interested if we can further improve its superior performance. We choose token size and sentence length (Tables 8 and 9), because we are interested in whether the benefits are intrinsic information are cumulative.

Table 6
RNNLM language models integrating two meta-information features (POS + X): results on CGN data under meta-information ‘known’ and ‘unknown’ conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
RNNME	112	21.3	112	21.3	38.7
POS	97	22.8	104	22.0	38.9
POS + SSS	94	23.0	102	22.0	39.1
POS + SL	99	22.6	106	22.1	38.8
POS + lemma	96	22.7	107	22.1	38.7
POS + T30	89	23.6	115	21.8	39.2
POS + TS	97	22.7	108	21.9	38.3

Table 7
RNNLM language models integrating two meta-information features (SSS + X): results on CGN data under meta-information ‘known’ and ‘unknown’ conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
RNNME	112	21.3	112	21.3	38.7
SSS	105	22.2	107	22.1	37.8
SSS + POS	94	23.0	102	22.0	39.1
SSS + T30	94	23.1	119	21.4	38.6
SSS + TS	106	22.2	107	22.0	38.3
SSS + SL	105	21.7	110	21.7	37.7
SSS + lemma	103	22.3	109	21.9	38.7

Table 8
RNNLM language models integrating two meta-information features (TS + X): results on CGN data under meta-information ‘known’ and ‘unknown’ conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
RNNME	112	21.3	112	21.3	38.7
TS	110	21.7	110	21.7	38.2
TS + POS	97	22.7	108	21.9	38.3
TS + T30	98	22.7	117	21.3	38.6
TS + SSS	106	22.2	107	22.0	38.3
TS + SL	110	21.8	110	21.8	38.2
TS + lemma	109	21.8	111	21.7	38.8

Examining the meta-information ‘known’ condition across all four tables yields an interesting insight. Adding a second feature improves performance consistently, but not without exception. Next, we turn to the meta-information unknown condition. Here, we see that an additive improvement when two data sources are combined cannot be taken for granted. In Table 6 we see that adding a second feature can occasionally boost performance, but does not consistently allow recovery of the performance lost when POS is ‘unknown’ (i.e., predicted), rather than ‘known’. In Table 7, we see that under the con-

Table 9
RNNLM language models integrating two meta-information features (SL + X): results on CGN data under meta-information 'known' and 'unknown' conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
RNNME	112	21.3	112	21.3	38.7
SL	109	21.9	109	21.9	37.9
SL + POS	99	22.6	106	22.1	38.8
SL + T30	101	22.7	118	21.2	38.2
SL + SSS	105	21.7	110	21.7	37.7
SL + TS	110	21.8	110	21.8	38.2
SL + lemma	107	22.1	110	21.9	38.6

Table 10
RNNLM language models integrating three or more meta-information features: results on CGN data under meta-information 'known' and 'unknown' conditions.

Model	Known		Unknown		
	PPL	WPA	PPL	WPA	WER
POS + SL + T30	88	23.5	112	21.8	38.7
SSS + lemma + TS	104	22.3	106	22.3	37.6
POS + SSS + T30	85	24.1	109	22.0	38.3
POS + lemma + T30	88	23.7	115	21.6	38.4
SSS + SL + TS	109	21.8	110	21.6	37.8
POS + SSS + SL + lemma + TS + T30	84	23.9	105	21.8	38.4

dition 'unknown' a second feature offers no improvement over using sss alone. In other words, the strong performance of sss is difficult to improve. In Tables 8 and 9, we notice a similar trend. The strong performance of these two intrinsic features is difficult to improve. Further combining them (TS + SL, which is the same as SL + TS) does not improve beyond the contribution of individual modalities.

These results support the conclusion that in order to successfully integrate multiple sources of meta-information, and be able to count on additive improvements, it is important that the underlying prediction be strong. The combination of different sorts of meta-information apparently reinforces the impact of meta-information prediction error, leading to less than satisfying results.

We close by noting that combinations of more than two meta-information sources also support this conclusion. We report results achieved when combining of three and more sources in Table 10. Again, under the known condition additive improvement is achieved. Under the unknown condition, combining multiple meta-information sources does not consistently yield an additive improvement. The picture that emerges is that it is relatively easy to get a basic boost in performance from integrating meta-information, but in order to exploit its full potential its prediction must be optimized.

4.3. WSJ experiment

4.3.1. Data

To further demonstrate the performance of the proposed models, in this section we carry out experiments on the English Wall Street Journal (WSJ) data set. We use the 100-best speech recognition list from the DARPA WSJ'92 and WSJ'93 data sets, as used by Mikolov et al. (2010, 2002). In the 100-best list set, 333 sentences are used as development data (DEV) for tuning the interpolation of language models score, acoustic model score and word insertion penalty. The rest, 465 sentences, are used for evaluation (EVAL). The oracle WER for development data and evaluation data are 6.1% and 9.5%, respectively. The training corpus contains 37 M words of running text from the NYT section of English Gigaword. The validation data set contains 186 K words. A held-out set of 230 K words is used for testing, especially for perplexity comparison. The vocabulary size of the training data is 192 K.

4.3.2. Experiment setup

For the WSJ data set, there is no human-annotated POS, lemma and sss meta-information available. For this reason, in order to obtain meta-information for use in our experiment, we use the Stanford CoreNLP tools (Manning et al., 2014) to generate POS and lemmas for the training data. We consider meta-information generated by a widely-available state-of-the-art tool to be the most natural alternative to hand-annotated meta-information, for reasons of reproducibility. In total, we have 36 different types of POS and 191 K lemmas. Recall that in the case of the WSJ data, we experiment with topic, rather than sss, as a type of discourse level meta-information. The topics are generated in the same way as described at the end of Section 3.2.2.

In the experiments on the WSJ data set in this section, all the models use 200 hidden neurons, 100 classes and one weight matrix with 1 billion elements that directly connect input to output. All the models are trained using Backpropagation Through Time (BPTT) with 5 steps. To integrate different meta-information for the WSJ data set, we use the same recipe as for the CGN data. Here, we also provide an additional comparison with an condition that uses and interpolation method of three models with different random initializations.

4.3.3. Experimental results

First we compare the models in terms of perplexity measured on the test data. The baseline model RNNME provides an improvement in perplexity over the Kneser–Ney 5-gram language model, lowering it from 174.5 to 108.3. A comparison of RNNME to the other models is shown in Fig. 3. By integrating different sources of meta-information, we achieve improvement over RNNME. Note that the greatest reduction in perplexity is achieved by Parts of Speech and lemmas. The performance when these types of meta-information are integrated using the RNNTLM (cf. 'POS' and 'lemma') is comparable to what is achieved when they are directly predicted using the CoreNLP toolkit (cf.

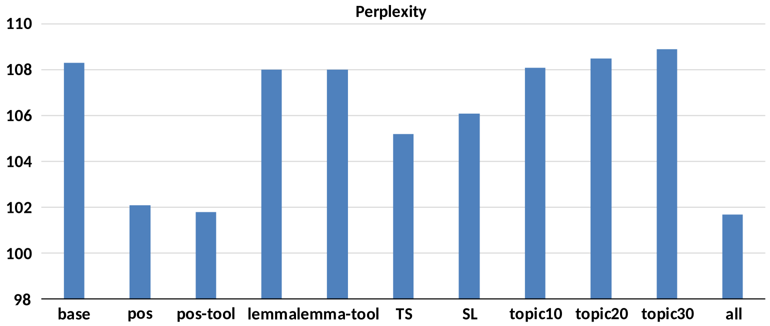

Fig. 3. Perplexity comparison of models using different meta-information on WSJ test data.

'POS-tool' and 'lemma-tool'). Another similar observation that can be made is the following: by integrating sentence length and token size information, the models achieve a perplexity reduction. A very slight improvement is achieved by using 10 topics. Using all types of meta-information (POS, TS, SL, topic10 and lemma) together, the final model can improve RNNME by 6% in terms of perplexity.

Next we turn to examine WER results, shown in Table 11. The WER results of the individual models as well as the interpolation of models with different random initializations are shown. We choose to carry out the comparison using an interpolation of three models, but also show interpolation of the baseline with 16 models (base × 16) for comparison. The results indicate that by using meta-information in addition to an interpolation strategy further improvement can be achieved over interpolation alone. Considering the individual meta-information types, we see that using POS can achieve the best result, which improves the baseline model (RNNME) by absolute 0.3 over the single model and 0.2 over the interpolation model. We see that POS delivers a performance improvement over the baseline. Using POS predicted from the CoreNLP tool ('POS-tool') performs slightly better than RNNTLM on the development data. Directly using lemmas predicted the tool ('lemma-tool') achieves exactly the same performance as RNNTLM. It is interesting to note that in terms of WER, POS shows different performance on the CGN data and WSJ data. With the CGN data, the perplexity improvement obtained by POS did not transfer to WER improvement. One possible reason is that CGN data has 281 different types of POS, which generates a much larger search space for POS than for the WSJ data set that only has 36 different POS. Also interesting is that the use of lemma information has relatively little impact on WER in the case of the WSJ data set, although it delivered a satisfying improvement in WER in the case of the CGN data set. We point out that this difference might reflect an underlying difference between English and Dutch. The relatively morphological richness of Dutch might lead to larger benefits from the use of lemma information. Next, we point out that the intrinsic meta-information (TS and SL) yields a small improvement, but does not make as large of a contribution as it did in the case of the CGN data. As with the CGN data, in Table 11 we see that our method of integrating topic information did not achieve a WER improvement over the baseline model. Finally, we remark that the combination of models is capable of yielding an improvement in WER, and, as illustrated by the last line in Table 11, the correct combination is capable of achieving a full 1% absolute improvement over the RNNME baseline. The bottom row of Table 11 shows the WER result using the interpolation of 16 language models including KN5GRAM, three RNNMEs, three POS integrated models, three TS integrated models, three SL integrated models and three models using the combination of POS, TS and SL information.

Table 11
(WER) comparison of models using different features on the WSJ data set, DEV and EVAL data. "$*\times 3$" means the interpolation of 3 models in rescoring. For 'POS' and 'lemma' the RNNTLM was used. For 'POS-tool' and 'lemma-tool' the meta-information for the test data was predicted directly using Stanford CoreNLP Tools

Model	Dev WER	Eval WER
KN5GRAM	12.2	17.2
Base	10.3	14.9
Base × 3	9.6	14.5
POS-tool	10.0	14.6
POS-tool × 3	9.4	14.2
POS	10.1	14.6
POS × 3	9.5	14.3
Lemma-tool	10.4	14.9
Lemma-tool × 3	9.6	14.5
Lemma	10.4	14.9
Lemma × 3	9.6	14.5
TS	10.2	14.8
TS × 3	9.6	14.3
SL	10.3	14.8
SL × 3	9.5	14.3
T10	10.4	14.9
T10 × 3	9.7	14.6
POS + TS + SL	10.0	14.4
(POS + TS + SL) × 3	9.4	14.0
Base × 16	9.3	14.0
All models except lemma and topic	9.3	13.9

5. Conclusions

In this paper, we have investigated the integration of meta-information into RNNLMs. We looked at three cases, the integration of word-level information using a Recurrent Neural Network Tandem Language Model (RNNTLM) architecture, the integration of discourse level information, and the integration of 'intrinsic' information, which can be derived directly without prediction.

The proposed methods were tested on two data sets. The first is the Spoken Dutch Corpus (CGN), which contain Dutch-language speech recordings, and the second is the Wall Street Journal, a well-known English-language data set. Our results based on experiments on these two data sets yield interesting insights. First, we noted that word-level meta-information yields a potential improvement, and it can be worthwhile using POS and lemma information, even if it must be predicted. However, there is a dependency of the contribution of the meta-data on the quality of the prediction, and in general performance under the meta-information 'known' condition was better than performance under the meta-data 'unknown' condition.

Second, we found that discourse-level information is capable of improving performance Adding information about social situational setting (SSS), recorded at the time of data capture, was shown to improve performance. Comparable levels of performance could be achieved when this information was predicted. Experiments with automatically created topics revealed that it is non-trivial to create discourse information that can yield a performance improvement when added to RNNLMS. Specifically, there is apparently a dependency between the amount of data available to train topics, and their ability to improve performance. In two different experiments, automatically derived topics did not improve performance.

Third, we have demonstrated the contribution that can be made by 'intrinsic' meta-information should not be overlooked. In fact, information sources such as token size and sentence length, which are trivial to derive, can make a contribution to RNNLMS that rivals that of meta-information that must be predicted.

Finally, our experiments with adding multiple sources of meta-information to RNNLMS point to the potential of additive improvement when sources are combined. If meta-information is reliable, combinations usually lead to improved performance, as witnessed by conditions involving intrinsic or 'known' meta-information. If meta-information must be predicted, and is, for this reason, less reliable, it becomes difficult to identify useful combinations. Our future work will be devoted to more robust prediction of meta-information, and combinations of meta-information.

The larger message of this paper is that RNNLMS offer an easy means of integrating meta-information into language models. Given the availability of meta-information, it is worthwhile attempting to exploit it in any given application scenario. Especially intrinsic information should be integrated into the model before attempting to exploit more costly or complex techniques.

References

Alexandrescu, A., Kirchhoff, K., 2006. Factored neural language models. In: Proceedings of the Human Language Technology Conference of the NAACL, pp. 1–4.

Antonio, J., Perez-Ortiz, Forcada, M.L., 2001. Part-of-speech tagging with recurrent neural networks. In: Proceedings of International Joint Conference of Neural Networks, pp. 1588–1592.

Bahl, L.R., Brown, P.F., de Souza, P.V., Mercer, R., 1989. A tree-based statistical language model for natural language speech recognition. IEEE Trans. Acoust., Speech Signal Process. 37 (7), 1001–1008.

Bellegarda, J.R., 1998. A multispan language modeling framework for large vocabulary speech recognition. IEEE Trans. Speech Audio Process. 6 (5), 456–467.

Bellegarda, J.R., Butzberger, J.W., Chow, Y.-L., Coccaro, N.B., Naik, D., 1996. A novel word clustering algorithm based on latent semantic analysis. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 172–175.

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.

Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning. Proceedings of International Conference on Machine Learning. ACM, pp. 41–48.

Bilmes, J.A., Kirchhoff, K., 2003. Factored language models and generalized parallel backoff. In: Proceedings of the Human Language Technology Conference of the NAACL, pp. 4–6.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022.

Bocchieri, E., Caseiro, D., Dimitriadis, D., 2011. Speech recognition modeling advances for mobile voice search. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4888–4891.

Botha, J.A., Blunsom, P., 2014. Compositional morphology for word representations and language modelling. In: Proceedings of the International Conference on Machine Learning.

Brown, P.F., de Souza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C., 1992. Class-based n-gram models of natural language. Comput. Linguist. 18, 467–479.

Chelba, C., 1997. A structured language model. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, pp. 498–500.

Chelba, C., Jelinek, F., 2000. Structured language modeling. Comp. Speech Lang. 14 (4), 283–332.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12, 2493–2537.

Cutting, D., Kupiec, J., Pedersen, J., Sibun, P., 1992. A practical part-of-speech tagger. In: Proceedings of The Third Conference On Applied Natural Language Processing, pp. 133–140.

Dean, T., Kanazawa, K., 1989. A model for reasoning about persistence and causation. Comput. Intell. 5 (3), 142–150.

Demuynck, K., 2001. Extracting, Modelling and Combining Information in Speech Recognition, Ph.D. thesis. KU Leuven, ESAT.

Demuynck, K., Laureys, T., Gillis, S., 2002. Automatic generation of phonetic transcriptions for large speech corpora. In: Proceedings of Interspeech, vol. I, pp. 333–336.

Demuynck, K., Roelens, J., Van Compernolle, D., Wambacq, P., 2008. SPRAAK: An open source SPeech Recognition and Automatic Annotation Kit. In: Proceedings of Interspeech, pp. 495–498.

Demuynck, K., Puurula, A., Van Compernolle, D., Wambacq, P., 2009. The ESAT 2008 system for N-Best Dutch speech recognition benchmark. In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 339–343.

Elman, J.L., 1993. Learning and development in neural networks: the importance of starting small. Cognition 48 (1), 71–99.

Emami, A., Jelinek, F., 2005. A neural syntactic language model. Mach. Learn. 60 (1–3), 195–227.

Gildea, D., Hofmann, T., 1999. Topic-based language models using EM. In: Proceedings of EUROSPEECH, pp. 2167–2170.

Heeman, P.A., 1999. POS tags and decision trees for language modeling. In: Proceedings of The Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 129–137.

Heidel, A., Chang, H.A., Lee, L.S., 2007. Language model adaptation using latent Dirichlet allocation and an efficient topic inference algorithm. In: Proceedings of Interspeech, pp. 2361–2364.

Jaynes, E.T., 1957. Information theory and statistical mechanics. Phys. Rev. Online Arch. (Prola) 106 (4), 620–630.

Kessens, J., van Leeuwen, D.A., 2007. N-Best: the Northern- and Southern-Dutch benchmark evaluation of speech recognition technology. In: Proceedings of Interspeech, pp. 1354–1357.

Luong, T., Socher, R., Manning, C., 2013. Better word representations with recursive neural networks for morphology. Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, pp. 104–113.

Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60.

Mesnil, G., He, X., Deng, L., Bengio, Y., 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Proceedings of Interspeech, pp. 3771–3775.

Mikolov, T., Zweig, G., 2012. Context dependent recurrent neural network language model. In: IEEE Workshop on Spoken Language Technology, pp. 234–239.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S., 2010. Recurrent neural network based language model. In: Proceedings of Interspeech, pp. 1045–1048.

Mikolov, T., Deoras, A., Kombrink, S., Burget, L., Ernock, J., 2011a. Empirical evaluation and combination of advanced language modeling techniques. In: Proceedings of Interspeech, pp. 605–608.

Mikolov, T., Deoras, A., Povey, D., Burget, L., Cernocký, J., 2011b. Strategies for training large scale neural network language models. In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 196–201.

Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S., 2011c. Extensions of recurrent neural network language model. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5528 –5531.

Mirowski, P., Chopra, S., Balakrishnan, S., Bangalore, S., 2010. Feature-rich continuous language models for speech recognition. In: Proceeding of IEEE Spoken Language Technology Workshop, pp. 241–246.

Morin, F., Bengio, Y., 2005. Hierarchical probabilistic neural network language model. In: AISTATS05, pp. 246–252.

Mousa, A.E.-D., Kuo, H.-K.J., Mangu, L., Soltau, H., 2013. Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8435–8439.

Murphy, K.P., 2002. Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis. University of California, Berkeley.

Ney, H., Essen, U., Kneser, R., 1994. On structuring probabilistic dependencies in stochastic language modelling. Comp. Speech Lang. 8.

Niesler, T., Woodland, P.C., 1996. Combination of word-based and category-based language models. In: Proceedings of International Conference on Spoken Language, vol. 1. pp. 220–223.

Niesler, T.R., Whittaker, E.W.D., Woodland, P.C., 1998. Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 177–180.

Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.P., Moortgat, M., Baayen, H., 2002. Experiences from the Spoken Dutch Corpus project. In: Araujo (Eds.), Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 340–347.

Pereira, F., Tishby, N., Lee, L., 1993. Distributional clustering of English words. In: Association for Computational Linguistics, pp. 183–190.

Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L., Roukos, S., 1992. Adaptive language modeling using minimum discriminant estimation. In: Proceedings of the workshop on Speech and Natural Language, pp. 103–106.

Putthividhya, D.P., Attias, H.T., Nagarajan, S., 2009. Independent factor topic models. Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp. 833–840.

Rosenfeld, R., 1996. A maximum entropy approach to adaptive statistical language modeling. Comp., Speech Lang. 10 (3), 187–228.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Shi, Y., Wiggers, P., Jonker, C.M., 2010. Language modelling with dynamic bayesian networks using conversation types and part of speech information. In: The 22nd Benelux Conference on Artificial Intelligence, pp. 154–161.

Shi, Y., Wiggers, P., Jonker, C.M., 2011. Combining topic specific language models. In: Proceedings of the International Conference on Text, Speech and Dialogue, pp. 99–106.

Shi, Y., Wiggers, P., Jonker, C.M., 2012. Towards recurrent neural networks language models with linguistic and contextual features. In: Proceedings of Interspeech, pp. 1664–1667.

Shi, Y., Wiggers, P., Jonker, C.M., 2013. Classifying the socio-situational settings of transcripts of spoken discourses. Speech Commun. 55 (10), 988–1002.

Shi, Y., Larson, M., Jonker, C.M., 2015. Recurrent neural network language model adaptation with curriculum learning. Comp. Speech Lang. 33 (1), 136–154. http://dx.doi.org/10.1016/j.csl.2014.11.004, ISSN 0885–2308.

Shi, Y., Yao, K., Chen, H., Pan, Y.-C., Hwang, M.-Y., Peng, B., 2015. Contextual spoken language understanding using recurrent neural networks. In: Proceedings of the IEEE International Conference on Accoustic Speech and Signal Processing.

Sigurd, B., Eeg-Olofsson, M., Van Weijer, J., 2004. Word length, sentence length and frequency—Zipf revisited. Stud. Linguist. 58 (1), 37–52.

Steinbiss, V., Tran, B.-H., Ney, H., 1994. Improvements in beam search. In: Proceedings of the International Conference on Spoken Language Processing, pp. 2143–2146.

Su, Y., 2011. Knowledge Integration Into Language Models: A Random Forest Approach. BiblioBazaar.

Ueberla, J.P., 1995. More efficient clustering of n-grams for statistical language modeling. In: Proceedings of EUROSPEECH, pp. 1257–1260.

van den Bosch, A., 2006. Scalable classification-based word prediction and confusible correction. Traitement Automatique des Langues 46 (2), 39–63.

Van Eynde, F., 2004. Part of speech tagging en lemmatisering van het corpus gesproken nederlands, Tech. rep. KU Leuven.

Wang, W., Harper, M.P., 2002. The superARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In: Proceedings of Conference of Empirical Methods in Natural Language Processing, pp. 238–247.

Wang, W., Vergyri, D., 2006. The use of word n-grams and parts of speech for hierarchical cluster language modeling. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. I–I.

Wiggers, P., Rothkrantz, L.J.M., 2006a. Dynamic bayesian networks for language modeling. In: Text and Speech and Dialogue, pp. 555–562.

Wiggers, P., Rothkrantz, L.J.M., 2006b. Topic-based language modeling with dynamic Bayesian networks. In: Proceedings of the Ninth International Conference on Spoken Language Processing, pp. 1866–1869.

Wiggers, P., Rothkrantz, L.J.M., 2007. Exploratory analysis of word use and sentence length in the spoken Dutch corpus. In: Proceedings of the International Conference on Text, Speech and Dialogue, pp. 366–373.

Wu, Y., Lu, X., Yamamoto, H., Matsuda, S., Hori, C., Kashioka, H., 2012. Factored language model based on recurrent neural network. In: Proceedings of International Conference of Computational Linguistics, pp. 2835–2850.

Xu, P., Jelinek, F., 2004. Random forests in language modeling. In: Proceedings of EMNLP, pp. 325–332.

Xu, P., Karakos, D., Khudanpur, S., 2009. Self-supervised discriminative training of statistical language models. In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 317–322.

Yamamoto, H., Sagisaka., Y., 1999. Multi-class composite n-gram based on connection direction. In: Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 533–536.

Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., Yu, D., 2013. Recurrent neural networks for language understanding. In: Proceedings of Interspeech, pp. 2524–2528.

Yule, G. Udny, 1939. On sentence-length as a statistical characteristic of style in prose: with application to two cases of disputed authorship. Biometrika 30 (3/4), 363–390.

Zipf, G.K., 1949. Human Behavior and the Principle of Least Effort. Addison-Wesley, Reading MA.