

Amsterdam University of Applied Sciences

Integrating meta-Information into recurrent neural network language models

Shi, Yangyang; Larson, Martha; Pelemans, Joris; Jonker, Catholijn M.; Wambacq, Patrick; Wiggers, Pascal; Demuynck, Kris

Published in:
Speech Communication

DOI:
[10.1016/j.specom.2015.06.006](https://doi.org/10.1016/j.specom.2015.06.006)

[Link to publication](#)

Citation for published version (APA):

Shi, Y., Larson, M., Pelemans, J., Jonker, C. M., Wambacq, P., Wiggers, P., & Demuynck, K. (2015). Integrating meta-Information into recurrent neural network language models. *Speech Communication*, 73(October), 64-80. <https://doi.org/10.1016/j.specom.2015.06.006>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please contact the library: <http://www.hva.nl/bibliotheek/contact/contactformulier/contact.html>, or send a letter to: University Library (Library of the University of Amsterdam and Amsterdam University of Applied Sciences), Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Amsterdam University of Applied Sciences

Integrating Meta-Information into Recurrent Neural Network Language Models

Wiggers, P.; Shi, Y.; Larson, M.; Pelemans, J.; Jonker, C.M.; Jacobs, Jasmien Decancq

Published in:
Speech Communication

[Link to publication](#)

Citation for published version (APA):

Wiggers, P., Shi, Y., Larson, M., Pelemans, J., Jonker, C. M., & Jacobs, J. D. (2015). Integrating Meta-Information into Recurrent Neural Network Language Models. *Speech Communication*, 73, 64-80.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please contact the library: <http://www.hva.nl/bibliotheek/contact/contactformulier/contact.html>, or send a letter to: University Library (Library of the University of Amsterdam and Amsterdam University of Applied Sciences), Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

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*

^b*ESAT Speech Group, Catholic University of Leuven*

^c*CREATE-IT Applied Research, Amsterdam University of Applied Sciences*

^d*Department of Electronics and Information Systems, University of Gent*

^e*HB12.290, Mekelweg 4, 2628CD Delft, The Netherlands. E-mail: yangyang.shi@tudelft.nl.*

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, lemma, social-situational setting, topic, token size, 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 sequence of symbols and only make use of the information from 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 the spoken words as *meta-*

8 *information*. Examples that are relevant to our investigation include word-level
9 meta-information such as Part of Speech (POS) or lemmas and discourse-level in-
10 formation such as the setting in which the speech is delivered (referred to as the
11 Social-Situational Setting) and topic.

12 Past efforts (Mirowski et al., 2010; Chelba, 1997; Shi et al., 2013; Bellegarda,
13 1998; Heidel et al., 2007) in language modeling have demonstrated that incorpo-
14 rating additional language-related information at different levels can improve the
15 performance of language models. Conventional n -gram language models (Brown
16 et al., 1992; Niesler et al., 1998; Heeman, 1999), however, offer relatively limited
17 possibilities for incorporating meta-information. In order to predict the next word
18 of a word sequence, a conventional n -gram language model relies solely on the
19 $n - 1$ words that precede it. This strategy is simple and robust, but is limited
20 in its ability to capture long distance dependencies between words and doesn't
21 generalize well from sparse data.

22 Recently, recurrent neural network language models (RNNLMs) (Mikolov et al.,
23 2010, 2011c) have demonstrated potential to address these shortcomings. The suc-
24 cess of RNNLMs can be attributed to two factors. First, RNNLMs map the discrete,
25 word-based vocabulary to a continuous space. This mapping makes it possible to
26 learn generalizations over word sequences that are not completely identical, thus
27 reducing the effect of data sparsity. Second, the recurrent loop in the RNNLM ar-
28 chitecture, which feeds the hidden layer back into the input layer at every time
29 step, constitutes a memory that serves to capture long-distance dependencies. In
30 this paper, we focus on a third advantage of RNNLMs that has received relatively
31 little attention in the literature. Incorporating meta-information into n -gram lan-
32 guage models is cumbersome. Generally, it is necessary to design specialized

33 architectures, to create hand-crafted models, or to train weighting parameters. In
34 contrast, integrating meta-information into RNNLMs just requires adding the extra
35 features to the input layer. Viewing recurrent neural networks as a set of logistic re-
36 gressions helps to make clear that adding extra information can be accomplished
37 elegantly: no special changes to the architecture of the model must be made in
38 order to accommodate the new information.

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

53 Our investigations cover three cases of meta-information. First, we investi-
54 gate *word-level linguistic information*, represented by Part of Speech (POS) tags
55 and lemmas. Previous work (Heeman, 1999; Wang and Vergyri, 2006) has estab-
56 lished that these sources enhance the performance of conventional language mod-
57 els. We confirm that RNNLMs also benefit from these sources. Second, we look at

58 *discourse-level information*, more specifically SSSs and topics. These sources of
59 meta-information are also known to improve conventional language models (Shi
60 et al., 2013; Gildea and Hofmann, 1999; Wiggers and Rothkrantz, 2006b). Here
61 again, we demonstrate their ability to improve the performance of RNNLMs. Fi-
62 nally, a third case concerns meta-information that can be considered *intrinsic*. In
63 other words, the information is inherent in words and word-sequences and does
64 not need to be inferred. Specifically, we investigate sentence length and token
65 size, two features that are readily available for exploitation, but which have been
66 largely overlooked in language modeling. It is difficult to identify a single factor
67 responsible for the lack of attention to intrinsic features in the literature. Most
68 likely, the oversight is due to the combination of the relatively large overhead re-
69 quired to integrate meta-information into conventional language models, already
70 mentioned above, and the a priori impression that intrinsic information is trivial.
71 When RNNLMs are used, the incorporation of extra information is straightforward
72 and elegant, and our experiments demonstrate that trivial information can be ex-
73 ploited to achieve performance gains.

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

83 used to predict topics for test data.

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

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

106 The rest of the paper is organized as follows. Section 2 discusses related work
107 on inferring meta-information and on previous methods that have exploited the

108 integration of meta-information into language models. In Section 3, we describe
109 our approach for incorporating meta-information into RNNLMs, including the RN-
110 NTLM architecture. Section 4 describe the experimental setup and present the
111 experimental results on the spoken Dutch data and English Wall Street Journal
112 data set. The final section provides conclusions and an outlook.

113 **2. Related Work**

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

118 *2.1. Meta-Information*

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

122 *2.1.1. Word-Level Meta-Information*

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

129 POS information is not an intrinsic property of a word, and for this reason, if it
130 is to be used in language modeling it must be predicted. Both the task of labeling

131 words with POS tags (Antonio et al., 2001; Cutting et al., 1992) and methods to
132 integrate POS with language models (Mirowski et al., 2010; Chelba, 1997) have
133 received considerable research attention. In this paper, the prediction of POS tags
134 as well as the integration of POS tags into language modeling is achieved by using
135 the proposed RNNTLMS.

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

140 Based on compositional morphological representations, Botha and Blunsom
141 (2014) proposed to integrate morphology into language modeling by factorizing
142 each word vector into its surface morphemes vectors. In (Mousa et al., 2013), the
143 mixture of words and morphemes along with their features were used as input to
144 Deep Neural Network language models. In (Luong et al., 2013), a context aware
145 word representation was constructed by applying Recursive Neural Networks on
146 a morphological binary tree.

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

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

160 2.1.2. Sentence Length (SL)

161 Sentence length is defined as the number of words in a sentence. It is a indica-
162 tor of discourse style and genre. This relationship was established even before the
163 advent of the Digital Age in the field of authorship attribution (Udny Yule, 1939).
164 Recent work observing the relationship between sentence length and genre in-
165 cludes Sigurd et al. (2004) and Wiggers and Rothkrantz (2007) for the spoken
166 Dutch data used in this work. In particular, sentence length distribution varies for
167 different conversation styles. For example, for spontaneous speech the average
168 sentence length is below 7. In spontaneous face-to-face conversations almost 25%
169 of the sentences contain only one word such as yes or no answers and interjections.
170 In contrast, the mean length of sentences in political discussion/debates/meetings
171 is 15, and in ceremonious speeches/sermons, it is 20. In n -gram language models
172 and conventional RNNLM, a sentence ending token is explicitly appended to each
173 sentence as a special word in the vocabulary. It helps to capture the kind of words
174 that are likely to occur at the end of a sentence. However, the exact sentence
175 length of a sentence is usually not modeled. An isolated exception may be Boc-
176 chieri et al. (2011), who demonstrated that combining separate language models,
177 each created for sentences of different lengths, improves recognition performance
178 in the domain of voice search. In our paper, we aim to exploit the benefits of
179 sentence length in a more general domain.

180 Applying sentence length information in language modeling can improve the

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

190 *2.1.3. Topic and Socio-Situational Settings*

191 Both the topic being spoken about and the situation in which language is used
192 (referred to here as Socio-situational Setting (SSS)), impact word distributions.
193 The topic is related to the subject under discussion by the speaker or speakers. In
194 contrast, the SSS is more of a proxy for the language register (style of speech),
195 which is influenced by the goal of the conversation, the relationship between
196 speakers and listeners, and the number of speakers and listeners involved. Certain
197 topics may be more typical of some SSSs than others, so in general it is not useful
198 to assume that the two are independent. The main distinction in the context of this
199 paper is that the SSS can be captured at the time of recording whereas regularities
200 that are discovered automatically in the data are considered to be topic related. In
201 research where topic is expected to mainly reflect the subject matter under discus-
202 sion, the topic models almost invariably differentiate between topics based on the
203 distribution of content words only, ignoring function words (Putthividhya et al.,
204 2009). In this work, we are interested in modeling underlying clusters in general,
205 and are agnostic if they are related to style or subject matter. Hence, all words are

206 allowed to contribute to the topic model.

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

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&d	Spontaneous telephone dialogues	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

213 2.2. *Language Models Integrating Information beyond Word Identity*

214 Previous research has established the usefulness of information that goes be-
215 yond the identity of words in improving language models. In this sub-section, we
216 survey some of the most successful work exploiting this information and explain
217 its relationship to our work.

218 Decision-tree-based language models (Bahl et al., 1989) are one of the ear-
219 lier language modeling methods that integrate meta-information with information
220 about word identity. For example, part-of-speech (POS) information can be in-
221 tegrated into language models by asking questions about the word history such
222 as, “Is the last word a verb?” (Heeman, 1999). In Su (2011), the Random For-
223 est Language models of Xu and Jelinek (2004) are extended with morphological,
224 prosodic, syntactic, and topic information.

225 Class-based language models (Brown et al., 1992) can be viewed as language
226 models that integrate meta-information. Since the quality of the class-based lan-
227 guage models depends on how the vocabulary is grouped into clusters, much pre-
228 vious research has been devoted to understanding the best way to cluster the vo-
229 cabulary (Brown et al., 1992; Ney et al., 1994; Ueberla, 1995; Pereira et al., 1993;
230 Bellegarda et al., 1996; Niesler and Woodland, 1996; Niesler et al., 1998; Ya-
231 mamoto and Sagisaka., 1999). Language models that group words according to
232 POS tag, allow easy integration of POS information with n -gram language mod-
233 els (Ney et al., 1994; Niesler et al., 1998). In this work, we show that automatic
234 determination of the categories yields improved performance over the original
235 POS categories, presumably because it allows control over the category size and
236 composition.

237 Structured language models (Chelba, 1997; Chelba and Jelinek, 2000) repre-

238 sent another important technique to exploit information beyond the word level.
239 These language models incorporate information concerning the syntactic struc-
240 ture of a language as well as the grammatical function of words in the form of
241 their POS class.

242 Some language models have integrated meta-information in an effort to better
243 encode information about long distance dependencies between words.

244 Language models incorporating latent topics (Bellegarda, 1998; HeideI et al.,
245 2007) are key examples. These models use a topical representation of the data
246 created by a method such as latent semantic analysis (Bellegarda, 1998) or latent
247 Dirichlet allocation (HeideI et al., 2007). In this paper, we also employ Latent
248 Dirichlet Allocation to construct a representation of documents that captures gen-
249 eralizations over topic. We then create topics by using k-means clustering.

250 Dynamic Bayesian Networks (DBNs) (Dean and Kanazawa, 1989; Murphy,
251 2002) offer a concise method to integrate additional features into a language
252 model. Syntactic information, semantic relationships and social background knowl-
253 edge can be simply specified as a variable into the network structure of the belief
254 network (Wiggers and Rothkrantz, 2006a; Shi et al., 2010, 2011). However, DBNs
255 are generalizations of n -gram language models, and as such share some of their
256 drawbacks. In particular, because they model exact sequences, they tend to suffer
257 in the face of sparse data.

258 Maximum entropy language models (Pietra et al., 1992; Rosenfeld, 1996)
259 are among the best existing methods for integrating additional information into
260 a language model. These models exploit the maximum entropy principle (Jaynes,
261 1957) in order to incorporate additional knowledge sources, which can be com-
262 pletely arbitrary in nature. In Rosenfeld (1996) maximum entropy language mod-

263 els using trigger and n -gram features are shown to achieve significant improve-
264 ment over n -gram language models in terms of perplexity and word error rate.
265 Maximum entropy language models can be viewed as a variety of neural net-
266 work language models which include no hidden layer. In this paper, we use the
267 maximum entropy extension of the RNNLM (RNNME) proposed by Mikolov et al.
268 (2011c), to incorporate meta-information into RNNLMs. The so called RNNME in-
269 cludes a direct connection between the input layer and the output layer effectively
270 incorporating a maximum entropy language model into the RNNLM architecture.

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

281 Factored language models proposed by Bilmes and Kirchhoff (2003) treat each
282 word as a vector of factors. In the work of Wu et al. (2012), the RNNLM is extended
283 to a factored RNNLM. However, in Wu et al. (2012), only word-level information
284 is used. In this paper, we investigate not only word-level information, but also
285 sentence-level and discourse-level information. Furthermore, we investigate the
286 incorporation of intrinsic information such as word and sentence length, which
287 initially gives the impression of being trivial, but actually has the ability to im-

288 prove language models. The usefulness of integrating topic information, derived
 289 via Latent Dirichlet Allocation, into RNNLMs has been studied by Mikolov and
 290 Zweig (2012). Here, we significantly expand on both Mikolov and Zweig (2012)
 291 and our own previous work on integrating linguistic and contextual information
 292 into RNNLMs (Shi et al., 2012). A full range of different types of meta-information
 293 is investigated. Further, we go beyond (Shi et al., 2012) in that we evaluate our
 294 models applied not only on the task of word prediction, but also on the task of N-
 295 best re-scoring. Lastly, we propose a recurrent neural network tandem language
 296 model (RNNTLM) which employs RNNLMs both for inferring meta-information
 297 and for predicting the probability of the next word.

298 **3. Recurrent Neural Network Tandem Language Models**

299 *3.1. Recurrent Neural Network Language Models*

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

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

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

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

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

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

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

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

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

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

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

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

325 In Mikolov et al. (2011c), a class-based RNNLM is proposed. A similar idea
 326 has also been investigated in Morin and Bengio (2005). The class-based RNNLM
 327 factorizes the output layer using classes. The classes are proportionally deter-
 328 mined according to the word frequency in the training data. For example, if we

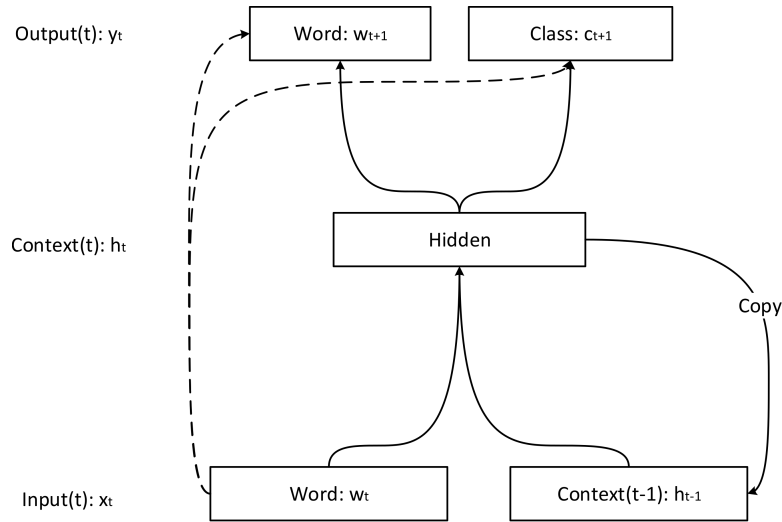


Figure 1: Class-based Maximum Entropy extension of RNNLMs. The dashed arrows represent the direct connection of n -gram features in the input to the output.

329 choose M classes, words that take up the top $\frac{1}{M}$ of the unigram distribution would
 330 be assigned to class 1. Using a class-based RNNLM, the probability of a word w_{t+1}
 331 at time $t + 1$ given its history $hist_{t+1}$ is calculated in the following way:

$$p(w_{t+1}|hist_t) = p(w_{t+1}|c_{t+1}, hist_t)p(c_{t+1}|hist_t), \quad (6)$$

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

341 3.2. *Recurrent Neural Network Tandem Language Models*

342 Our approach to integrate meta-information into RNNLMs consists of mod-
343 els containing two parts, one part uses a recurrent neural network for predicting
344 the meta-information, and the other part integrates the predicted meta-information
345 into RNNLMs. To predict multiple types of meta-information, several individual
346 recurrent neural networks are used. Recently (Shi et al., 2015; Collobert et al.,
347 2011) use multi-task learning to use one model to predict different types of in-
348 formation, but exploration of a single model to predict different types of meta-
349 information lies beyond the scope of the current paper.

350 Different types of meta-information predictors are needed to extract the vari-
351 ous types of meta-information used in this study. Meta-information such as token
352 size and sentence length, is ‘intrinsic’, meaning that it can be derived directly by
353 inspecting the data. Both the token size and sentence length are encoded using the
354 1-of-N method. In the testing, the unseen token size or sequence length informa-
355 tion is ignored by triggering a zero vector. All the words in the same sentence bear
356 the same sentence length information. Sentence length correlates with the topic or
357 the social situational settings of current sentence. Using such an encoding method,
358 we actually cluster the sentences according to sentence length.

359 However, to obtain the word-level information (POS, lemma) and the discourse-
360 level information (Socio-Situational Settings and topics) for the test data, we need
361 the aid of a meta-information predictor. In the following subsections, we discuss
362 the two cases in turn.

363 3.2.1. *Integrating Word-Level Meta-Information*

364 Word-level meta-information is predicted using the history of the current word.
365 In order to incorporate word-level meta-information, we use the recurrent neu-

366 ral network tandem language model (RNNTLM) architecture that is illustrated in
 367 Fig 2. The meta-information prediction component is an RNNLM as well. In
 368 order to predict meta-information $m(t)$ for the current word $w(t)$, the previous
 369 meta-information $m(t - 1)$, the current word $w(t)$ and the copied hidden layer
 370 $h_m(t - 1)$ are fed into the network.

$$x(t) = [w(t)^T m_1(t - 1)^T \dots m_p(t - 1)^T h(t - 1)^T]^T, \quad (7)$$

371 where p is the number of types of meta-information. The word vector $w(t)$ and all
 372 meta-information vectors $m_{1\dots p}(t - 1)$ are represented using a 1-of-N encoding.
 373 Because the maximum entropy extension is used, as is shown in equation (4), the
 374 previous $n - 1$ words, the current word, the previous $n - 1$ meta-information vec-
 375 tors and the current meta-information vector are directly connected to the meta-
 376 information output layer. The sequences of words and the sequence of meta-
 377 information vectors are encoded as large hash based vectors using encoding 1-
 378 of-N. In this paper, both word sequence and meta-information sequence are repre-
 379 sented by hash vectors with 1 billion elements. Note that the input to the maximum
 380 entropy extension part is different from the input to the RNNLMs part. The input to
 381 the maximum entropy part is much larger than the input to the RNNLMs. For con-
 382 venience, in Fig 2, we indicate the maximum entropy extension by using dashed
 383 lines that directly connect the input layer of the RNNLMs to the output layer of the
 384 RNNLMs. The maximum entropy extension integrates additional regular n -gram
 385 features into RNNLMs. The fixed and limited length of n -grams turns out to be
 386 complementary to the variable-length history generalizations learned by the re-
 387 current connection of the RNN, allowing the RNNME to capture local regularities.
 388 In the output layer of the meta-information predictor, the meta-information $m^*(t)$
 389 that obtains the highest probability is selected and encoded in a 1-of-N represen-

390 tation, which is fed to the RNNLM:

$$m^*(t) = \arg \max_{m(t)} p(m(t)|w(t), m(t-1), \text{hist}(t-1)). \quad (8)$$

391 When the meta-information is unknown, the current predicted meta-information
 392 $m^*(t)$ is copied to the input of the meta-information predictor in order to predict
 393 next meta-information $m^*(t+1)$.

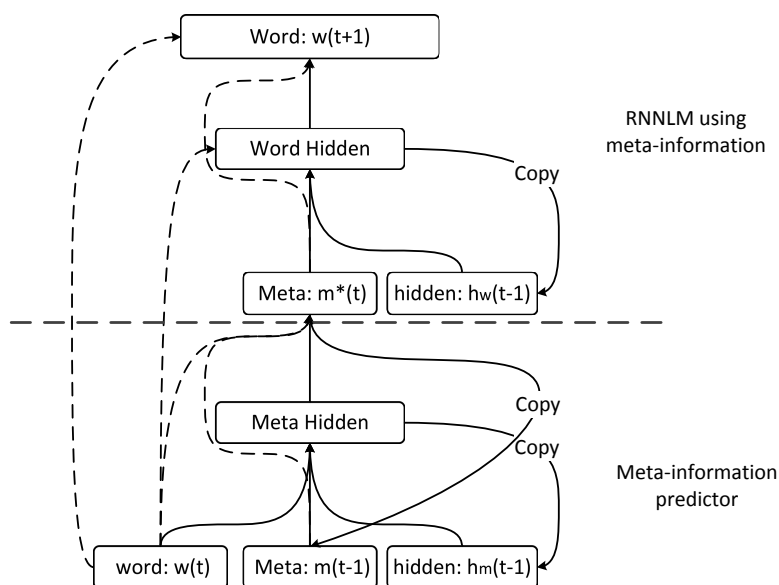


Figure 2: Recurrent neural network tandem language models integrating word-level information. The RNN below the dashed line predicts meta-information on the word level. The RNN above the dashed line incorporates the meta-information. The dashed arrows represent the direct connection of n -gram features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

394 As shown in Fig 2, the part above the horizontal dashed line is also a recurrent
 395 neural network. It uses the current word $w(t)$, and the predicted meta-information
 396 for the current word $m^*(t)$ and the copied hidden layer $h_w(t-1)$, to predict the
 397 next word $w(t+1)$. In the proposed RNNTLM, the structures of the two recurrent
 398 neural networks are basically the same; they differ only in their input and output.

399 3.2.2. *Integrating Discourse-Level Meta-Information*

400 Discourse-level information is predicted using a recurrent neural network as
401 well, when the information is not available in testing.

402 Because the RNNLM is applied within an N-Best rescoring framework, seg-
403 ment information is available at the moment the language model is applied. This
404 information has been generated by the speech recognition system that produced
405 the N-Best lists. By looking at the full segment instead of only at the preceding
406 words (cf. Fig 1), a better prediction of the discourse-level information can be
407 obtained.

408 We test under known and unknown conditions. Under the ‘known’ condi-
409 tion, the information about the correct discourse-level meta-information category
410 is available at test time. Under the ‘unknown’ condition, the information must
411 be predicted. In order to predict discourse-level meta-information, we train one
412 sub-domain-specific RNNLM for each SSS or topic (also referred to as a compo-
413 nent). This model is trained using the curriculum learning method for training
414 domain-adapted RNNLMs. Our previous work has demonstrated the effectiveness
415 of this method for creating RNNLMs for a heterogeneous domain that is com-
416 posed of a number of sub-domains (Shi et al., 2014). Curriculum learning (Ben-
417 gio et al., 2009; Elman, 1993) makes use of the fact that neural networks are
418 sensitive to the order in which data is presented to them during training. By pre-
419 senting the RNNLM first with general domain data and only later in the training
420 phase with sub-domain data, we create models which emphasize the patterns in
421 the sub-domain data. Curriculum learning can be regarded as a form of implicit
422 interpolation between a domain model and a sub-domain model. It achieves the
423 same goal as conventional linear interpolation, but does so with a single, contin-

424 ously trained model that dispenses with the need to explicitly train weights of
 425 individual sub-models. Discourse-level labels, which are predicted at the segment
 426 level, are duplicated for each word.

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

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

$$c(s) = \arg \max_k p(k|s) = \arg \max_k \frac{p_k(s)p(k)}{p(s)} = \arg \max_k p_k(s)p(k), \quad (9)$$

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

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

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

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

442 The second type of discourse-level information considered in this work are
 443 topics. The topics are derived automatically from the training data using Latent
 444 Dirichlet Allocation (LDA) (Blei et al., 2003) in conjunction with k -means clus-
 445 tering. LDA is a probabilistic model that describes the generation process of doc-

446 uments. A document is considered to be a mixture of underlying topics that give
447 rise to the words it contains. LDA applies a bag-of-words strategy, allowing each
448 document to be represented as a latent topic vector whose components reflect the
449 relative contributions of the individual latent topics. We choose to make use of
450 LDA since it represents the state of the art in topic representations. We construct
451 latent topic representations by considering each segment to be a document. We
452 then apply k -means clustering in order to cluster the data. The result is a set of
453 topic clusters. Each word in a segment bears the topic label of the cluster the
454 segment belongs to.

455 **4. Experiments and Results**

456 In this section, we describe the setup used to carry out our experiments on
457 the CGN data set (Dutch) and the WSJ (English) data set, and present the results
458 of our experimental investigation. We start out by providing some overarching
459 information concerning the two sets of experiments. The goal of the experiments
460 is to investigate the added value of adding meta-information to RNNLMs, and to
461 gain insight into which meta-information is most useful in which situations. As
462 mentioned above, we are interested in two scenarios meta-information ‘known’,
463 in which the meta-information is available at test time, and ‘unknown’, in which
464 the meta-information must be predicted. This comparison allows us to understand
465 the impact of meta-information prediction errors on our RNNLM language mod-
466 els. For intrinsic features, token size and sentence length, are the same for both
467 conditions, since they are trivial to compute. For word-level and discourse-level
468 features, the CGN data set offers the possibility to directly compare ‘known’ and
469 ‘unknown’, since the data set includes ground-truth for POS, lemma, and SSS.

470 The WSJ data set does not include similar ground truth. For this reason, we use
471 the Stanford CoreNLP tools, as a high quality method to predict word-level meta-
472 information, POS and lemma. We use this meta-information directly at predicted
473 meta-information to train our RNNTLM, and also We use this meta-information
474 directly at test time, to emulate the ‘known’ condition, and we use it as training
475 data for the ‘unknown’ condition that uses the RNNTLM approach to predict meta-
476 information. For both the CGN and WSJ data sets, we use topics, automatically
477 discovered by the process described below, in order to experiment with discourse-
478 level information. In the case of CGN, these topics provide us a contrast with SSS
479 meta-information, provided with the data set.

480 *4.1. Evaluation Metrics*

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

494 4.2. CGN Experiment

495 4.2.1. Data

496 In this section, the language model training and test data comes from the Spo-
497 ken Dutch Corpus (Corpus Gesproken Nederlands, CGN) (Oostdijk et al., 2002),
498 which contains recordings of standard Dutch spoken by adults in the Netherlands
499 and Flanders in a variety of language usage settings. As shown in Table 1, the
500 entire corpus contains nearly 9 million words divided into 14 components. We
501 used the component as a proxy for the socio-situational setting. Each component
502 is further divided into segments that contain one or more sentences. Segments
503 may be as large as 1,000 words.

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, l, 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

504 Components *a* to *h* contain dialogues or multilogues and components *i* to *o*
505 contain monologues. Our experiments are carried out on a test set that contains
506 10% of the data randomly selected from components *h*, *g*, *n* and *o*. The choice of
507 these components was made by practical considerations, which included the need
508 to exclude the data used to train the acoustic models for the speech recognition
509 system that generated the N-best list (further described in Section 4.2.4).

510 In total the test set contains 974K running words and 149 segments. For lan-
511 guage model training, 80% of the CGN data, mutually exclusive from the test set,
512 was used. Another mutually exclusive set of 10% of the data was used for valida-
513 tion. The details of the training data size for each part of the speech recognition
514 system is described in Table. 2.

515 4.2.2. *Part-Of-Speech And Lemma Prediction*

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

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

535 the training of the lemma predictor, the classes are determined according to the
536 lemma frequency in the training data. Prediction proceeds in two steps. First, we
537 predict the class using Eq. 11.

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

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

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

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

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

553 4.2.3. Socio-Situational Setting and Topic Prediction

554 Both SSS and topic we mentioned in this paper are simply ways of dividing the
555 data set into subsets that can be helpful. The SSS was collected when the data was

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&d	e	f	g	h	i	j	k	l	n	o
a	81						19						
b		98			2								
c&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

556 recorded (see Section 4.2.1). The topic information is derived by LDA and clus-
557 tering was described at the end of Section 3.2.2. Note that the notion of topic used
558 in this paper may also contain other information (e.g., style). We refer the output
559 from LDA and clustering algorithm as topic because these algorithms are generally
560 used to find topics. Prediction of SSS and topic for the N-best lists returned by the
561 recognizer is carried out using the RNNLM-based prediction method described in
562 Figure 2. It is important to note that this method achieves good performance in
563 predicting SSS. The average accuracy on the test data (that covers all of the com-
564 ponents in the CGN corpus) is 76.2%. As shown in Table 3, nine out of thirteen
565 components achieve above 95% accuracy. The performance for the components
566 that are used for N-best lists is: 98%, 96%, 93% and 100% for components *h*, *g*,
567 *n* and *o*, respectively. The main challenge for our SSS classifier comes from the
568 components *c&d*, which achieves the lowest accuracy.

569 4.2.4. *Generating The N-Best List*

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

578 The acoustic models employ 49 three-state acoustic units (46 phones, silence,
579 garbage and speaker noise) and one single-state phone (short schwa), which are
580 modeled using SPRAAK’s default tied Gaussian approach. Under this approach,

581 the density function for each of the 4k cross-word context-dependent tied states is
582 modeled as a mixture of an arbitrary subset of Gaussians drawn from a global pool
583 of 50k Gaussians. The mixtures use on average 180 Gaussians to model a 36 di-
584 mensional observation vector of MIDA features (Demuynck, 2001). These were
585 obtained by means of a mutual information based discriminant linear transform
586 (MIDA) on vocal-tract length normalized (VTLN) and mean-normalized MEL-
587 scale spectral features and their first and second order time derivatives. The acous-
588 tic models are trained on Broadcast News (components f, i, j, k, l in Table 1) and
589 Conversational Telephone Speech (components c, d in Table 1).

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

604 Since Dutch compounds are always written as a single word, the word recog-
605 nition results are post-processed for compounding. Two subsequent words are

606 replaced by their compound if the following criteria are met: 1) the words are
607 longer than 3 letters, 2) the words are not very rare, 3) the unigram count of the
608 compound is higher than the bigram count of the individual words. This approach
609 effectively extends the 400k lexicon to a 6M lexicon.

610 The main parameters of the system control hypothesis pruning and combining
611 the language model and the acoustic models. To combine the model scores, we
612 employ our standard way of handling this problem (Demuynck, 2001), by having
613 a LM scaling factor and a word startup cost. Beam search pruning was applied
614 to control the amount of hypotheses in the search space (Steinbiss et al., 1994):
615 a threshold indicates how much the score of a hypothesis can drop below the
616 score of the most likely hypothesis; if most hypotheses have a similar score, a
617 beam width parameter is applied to indicate how many hypotheses can be retained,
618 keeping only the best ones.

619 Adopting the pruning parameters that yield recognition in real time, we create
620 a lattice with the most likely word sequence hypotheses for each speaker turn in
621 each component. Using SRI's lattice tool, each lattice is converted into an N-best
622 list containing the 10000 best sentences, disregarding filler words and silences.

623 *4.2.5. Re-scoring The N-Best List With The RNNLMs Integrating Meta-Information*

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

629 4.2.6. *Experimental Results*

630 In this section, we present the results obtained with our proposed approach for
631 integrating meta-information into RNNLMs. We compare our models to two base-
632 lines, KN5GRAM, which is a conventional Kneser-Ney 5-gram language model
633 and also RNNME, which is the same RNNLM that we use in our approach, except
634 for the fact that it does not integrate any meta-information. As shown in Table 4,
635 when applied to the task of N-best restoring, the conventional Kneser-Ney 5-gram
636 language model achieves a WER of 40.1 on our CGN data set, and is outperformed
637 by the RNNME language model, which achieves a WER of 38.7. The experiments
638 investigate two scenarios: the ‘known’ condition, which is an oracle condition
639 under which the information is known at test time, and the ‘unknown’ condition,
640 under which the RNNTLM architecture in Fig 2 is used to predict the information
641 at test time. Note that these two conditions are the same for the baselines, which
642 do not integrate any meta-information.

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

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

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

654 the theoretical performance achievable under the oracle condition.

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

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

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

679 of topics are both aspects that must be taken into consideration when integrating
 680 topic as meta-information into an RNNLM.

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

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

687 Now, we turn to the topic of integrating ‘intrinsic’ meta-information into RNNLMs,
 688 i.e., token size and sentence length (cf. the lines labeled ‘TS’ and ‘SL’ in Table 4).
 689 Recall that intrinsic meta-information is particularly interesting since its use has
 690 been largely overlooked in the literature on conventional language models. It is
 691 ‘free’ information in the sense that it can be derived directly, without the need
 692 for prediction. Note that because intrinsic information uses counts of letters in
 693 words (TS) and of words in the utterance being rescored (SL), the results of the
 694 ‘known’ and the ‘unknown’ condition are the same. Both with respect to WPA and

695 with respect to WER, intrinsic meta-information is able to achieve performance
696 improvement over the RNNME baseline. The performance is slightly better in the
697 case of SL than in the case of TS. It is particularly striking that the WER falls by
698 0.5 absolute in the case of TS (38.7 to 38.5), and by 0.8 absolute in the case of
699 SL (38.7 to 37.9). In summary, these results suggest that intrinsic information,
700 although trivial to derive, should not be considered trivial when it comes to in-
701 tegrating meta-data into RNNLMs. Instead, this sort of ‘free’ information should
702 be exploited. It is capable of yielding a performance improvement of the same
703 magnitude of the one attainable by more costly methods that require the training
704 of a meta-information predictor.

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

705 Finally, we turn to the experiments that integrating multiple feature simultane-
706 ously into RNNLMs. We first consider experimental results obtained when adding
707 another feature, to a selection of the conditions in Table 4. Results are reported

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

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

708 separately for easy comparison in Tables 6, 7, 8 and 9. We choose POS (Tab. 6)
709 as a representative word-level feature, because we are interested in whether addi-
710 tional features can close the gap between the ‘known’ and ‘unknown’ condition.
711 We choose SSS (Tab. 7) as a representative discourse-level feature, because we are
712 interested if we can further improve its superior performance. We choose token
713 size and sentence length (Tables 8 and 9), because we are interested in whether
714 the benefits are intrinsic information are cumulative.

715 Examining the meta-information ‘known’ condition across all four tables yields
716 an interesting insight. Adding a second feature improves performance consis-
717 tently, but not without exception. Next, we turn to the meta-information unknown
718 condition. Here, we see that an additive improvement when two data sources are
719 combined cannot be taken for granted. In Tab. 6 we see that adding a second
720 feature can occasionally boost performance, but does not consistently allow re-
721 covery of the performance lost when POS is ‘unknown’ (i.e., predicted), rather
722 than ‘known’. In Tab. 7, we see that under the condition ‘unknown’ a second
723 feature offers no improvement over using SSS alone. In other words, the strong
724 performance of SSS is difficult to improve. In Tables 8 and 9, we notice a similar
725 trend. The strong performance of these two intrinsic features is difficult to im-
726 prove. Further combining them (TS +SL, which is the same as SL +TS) does not
727 improve beyond the contribution of individual modalities.

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

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

741 *4.3. WSJ Experiment*

742 *4.3.1. Data*

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

755 4.3.2. *Experiment Setup*

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

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

773 4.3.3. *Experimental Results*

774 First we compare the models in terms of perplexity measured on the test
775 data. The baseline model RNNME provides an improvement in perplexity over
776 the Kneser-Ney 5-gram language model, lowering it from 174.5 to 108.3. A com-
777 parison of RNNME to the other models is shown in Fig. 3. By integrating different
778 sources meta-information, we achieve improvement over RNNME. Note that the
779 greets reduction in perplexity is achieved by Parts of Speech and lemmas. The

780 performance when these types of meta-information are integrated using the RN-
 781 NTLM (cf. ‘POS’ and ‘lemma’) is comparable to what is achieved when they are
 782 directly prediction using the CoreNLP toolkit (cf. ‘POS-tool’ and ‘lemma-tool’).
 783 Another similar observation that can be made is the following: by integrating
 784 sentence length and token size information, the models achieve a perplexity re-
 785 duction. A very slight improvement is achieved by using 10 topics. Using all
 786 types of meta-information (POS, TS, SL, topic10 and lemma) together, the final
 787 model can improve RNNME by 6% in terms of perplexity.

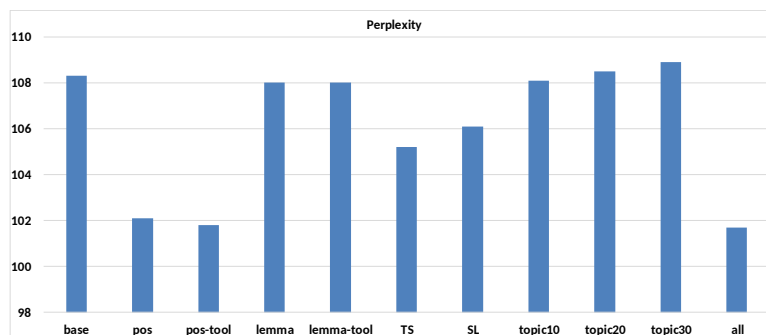


Figure 3: Perplexity comparison of models using different meta-information on WSJ test data.

788 Next we turn to examine WER results, shown in Table 11. The WER results of
 789 the individual models as well as the interpolation of models with different random
 790 initialization are shown. We choose to carry out the comparison using an interpo-
 791 lation of three models, but also show interpolation of the baseline with 16 models
 792 (base \times 16) for comparison. The results indicate that by using meta-information
 793 in addition to an interpolation strategy can achieves further improvement over in-
 794 terpolation alone. Considering the individual meta-information, using POS can
 795 achieve best result that improves the baseline model (RNNME) by absolute 0.3
 796 on single model and 0.2 on the interpolation model. We see that POS delivers

797 a performance improvement over the baseline. Using POS predicted from the
798 CoreNLP tool ('POS-tool') performs slightly better than RNNTLM on the devel-
799 oping data. Directly using lemmas predicted the tool ('lemma-tool') achieves
800 exactly the same performance as RNNTLM. It is interesting to note that in terms
801 of WER, POS shows different performance in CGN data and WSJ data. With the
802 CGN data, the perplexity improvement obtained by POS did not transfer to WER
803 improvement. One possible reason is that CGN data has 281 different types of POS
804 that generates much larger search space for POS than in the WSJ data set that only
805 has 36 different POS. Also interesting is that the use of lemma information has rel-
806 atively little impact on WER in the case of the WSJ data set, although it delivered a
807 satisfying improvement in WER in the case of the CGN data set. We point out that
808 this difference might reflect an underlying difference between English and Dutch.
809 The relatively morphological richness of Dutch might lead to larger benefits of the
810 use of lemma information. Next, we point out that the intrinsic meta-information
811 (TS and SL) yields a small improvement, but does not make as large of a contribu-
812 tion as it did in the case of the CGN data. As with the CGN data, in Table 11 we see
813 that our method of integrating topic information did not achieve a WER improve-
814 ment over baseline model. Finally, we remark that the combination of models is
815 capable of yielding an improvement in WER, and, as illustrated by the last line
816 in Table 11, the correct combination is capable of achieving a full 1% absolute
817 improvement over the RNNME baseline. The bottom row of Table. 11 shows the
818 WER result using the interpolation of 16 language models including KN5GRAM,
819 three RNNMES, three POS integrated models, three TS integrated models, three
820 SL integrated models and three models using the combination of POS, TS and SL
821 information.

Table 11: (WER) comparison of models using different features on the WSJ data set, DEV and EVAL data. “*×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

822 **5. Conclusions**

823 In this paper, we have investigated the integration of meta-information into
824 RNNLMs. We looked at three cases, the integration of word-level information
825 using a Recurrent Neural Network Tandem Language Model (RNNTLM) architec-
826 ture, the integration of discourse level information, and the integration of ‘intrin-
827 sic’ information, which can be derived directly without prediction.

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

838 Second, we found that discourse-level information is capable of improving
839 performance Adding information about Social Situational Setting (SSS), recorded
840 at the time of data capture, was shown to improve performance. Comparable lev-
841 els of performance could be achieved when this information was predicted. Ex-
842 periments with automatically created topics revealed that it is non-trivial to create
843 discourse information that can yield a performance improvement when added to
844 RNNLMs. Specifically, there is apparently a dependency between the amount of
845 data available to train topics, and their ability to improve performance. In two
846 different experiments, automatically derived topics did not improve performance.

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

851 Finally, our experiments with adding multiple sources of meta-information to
852 RNNLMs point to the potential of additive improvement when sources are com-
853 bined. If meta-information is reliable, combinations usually lead to improved
854 performance, as witnessed by conditions involving intrinsic or ‘known’ meta-
855 information. If meta-information must be predicted, and is, for this reason, less
856 reliable, it becomes difficult to identify useful combinations. Our future work will
857 be devoted to more robust prediction of meta-information, and combinations of
858 meta-information.

859 The larger message of this paper is that RNNLMs offer an easy means of in-
860 tegrating meta-information into language models. Given the availability of meta-
861 information, it is worthwhile attempting to exploit it in any given application sce-
862 nario. Especially intrinsic information should be integrated into the model before
863 attempting to exploit more costly or complex techniques.

864 **References**

865 Alexandrescu, A., Kirchhoff, K., 2006. Factored neural language models. In: Pro-
866 ceedings of the Human Language Technology Conference of the NAACL. pp.
867 1–4.

868 Antonio, J., Perez-Ortiz, Forcada, M. L., 2001. Part-of-speech Tagging with Re-
869 current Neural Networks. In: Proceedings of International Joint Conference of
870 Neural Networks. pp. 1588–1592.

871 Bahl, L. R., Brown, P. F., de Souza, P. V., Mercer, R., 1989. A tree-based statistical
872 language model for natural language speech recognition. *IEEE Transactions on*
873 *Acoustics, Speech and Signal Processing* 37 (7), 1001–1008.

874 Bellegarda, J. R., 1998. A multispans language modeling framework for large vo-
875 cabulary speech recognition. *IEEE Transactions on Speech and Audio Process-*
876 *ing* 6 (5), 456–467.

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

881 Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., 2003. A neural probabilistic
882 language model. *Journal of Machine Learning Research* 3, 1137–1155.

883 Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning.
884 In: Proceedings of International Conference on Machine Learning. ACM, pp.
885 41–48.

- 886 Bilmes, J. A., Kirchhoff, K., 2003. Factored language models and generalized parallel
887 backoff. In: Proceedings of the Conference of the North American Chapter
888 of the Association for Computational Linguistics on Human Language Technology.
889 pp. 4–6.
- 890 Blei, D. M., Ng, A. Y., Jordan, M. I., 2003. Latent Dirichlet allocation. *Journal of*
891 *Machine Learning Research* 3, 993–1022.
- 892 Bocchieri, E., Caseiro, D., Dimitriadis, D., 2011. Speech recognition modeling
893 advances for mobile voice search. In: Proceedings of IEEE International Conference
894 on Acoustics, Speech and Signal Processing. pp. 4888–4891.
- 895 Botha, J. A., Blunsom, P., 2014. Compositional Morphology for Word Representations
896 and Language Modelling. In: Proceedings of the International Conference on
897 Machine Learning.
- 898 Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., Lai, J. C., 1992.
899 Class-based n-gram models of natural language. *Computational Linguistics* 18,
900 467–479.
- 901 Chelba, C., 1997. A structured language model. In: Association for Computational
902 Linguistics. pp. 498–500.
- 903 Chelba, C., Jelinek, F., 2000. Structured language modeling. *Computer Speech &*
904 *Language* 14 (4), 283–332.
- 905 Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.,
906 2011. Natural language processing (almost) from scratch. *Journal of Machine*
907 *Learning Research* 12, 2493–2537.

- 908 Cutting, D., Kupiec, J., Pedersen, J., Sibun, P., 1992. A practical part-of-speech
909 tagger. In: Proceedings of The Third Conference On Applied Natural Language
910 Processing. pp. 133–140.
- 911 Dean, T., Kanazawa, K., 1989. A model for reasoning about persistence and cau-
912 sation. *Computational Intelligence* 5 (3), 142–150.
- 913 Demuynck, K., 2001. Extracting, modelling and combining information in speech
914 recognition. Ph.D. thesis, K.U.Leuven ESAT.
- 915 Demuynck, K., Laureys, T., Gillis, S., 2002. Automatic generation of phonetic
916 transcriptions for large speech corpora. In: Proceedings of Interspeech. Vol. I.
917 pp. 333–336.
- 918 Demuynck, K., Puurula, A., Van Compernelle, D., Wambacq, P., 2009. The ESAT
919 2008 system for N-Best Dutch speech recognition benchmark. In: IEEE work-
920 shop on automatic speech recognition and understanding. pp. 339–343.
- 921 Demuynck, K., Roelens, J., Van Compernelle, D., Wambacq, P., 2008. SPRAAK:
922 An open source SPeech Recognition and Automatic Annotation Kit. In: Pro-
923 ceedings of Interspeech. pp. 495–498.
- 924 Elman, J. L., 1993. Learning and development in neural networks: the importance
925 of starting small. *Cognition* 48 (1), 71 – 99.
- 926 Emami, A., Jelinek, F., 2005. A neural syntactic language model. *Machine Learn-*
927 *ing* 60 (1-3), 195–227.
- 928 Gildea, D., Hofmann, T., 1999. Topic-based language models using EM. In: Pro-
929 ceedings of EUROSPEECH. pp. 2167–2170.

- 930 Heeman, P. A., 1999. Pos tags and decision trees for language modeling. In: Pro-
931 ceedings of The Joint SIGDAT Conference on Empirical Methods in Natural
932 Language Processing and Very Large Corpora. pp. 129–137.
- 933 Heidel, A., Chang, H. A., Lee, L. S., 2007. Language model adaptation using
934 latent dirichlet allocation and an efficient topic inference algorithm. In: Pro-
935 ceedings of Interspeech. pp. 2361–2364.
- 936 Jaynes, E. T., May 1957. Information Theory and Statistical Mechanics. Physical
937 Review Online Archive (Prola) 106 (4), 620–630.
- 938 Kessens, J., van Leeuwen, D. A., 2007. N-Best: the Northern- and Southern-
939 Dutch benchmark evaluation of speech recognition technology. In: Proceedings
940 of Interspeech. pp. 1354–1357.
- 941 Luong, T., Socher, R., Manning, C., 2013. Better word representations with recur-
942 sive neural networks for morphology. In: Proceedings of the Seventeenth Con-
943 ference on Computational Natural Language Learning. Association for Com-
944 putational Linguistics, pp. 104–113.
- 945 Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., McClosky, D.,
946 2014. The Stanford CoreNLP natural language processing toolkit. In: Proceed-
947 ings of 52nd Annual Meeting of the Association for Computational Linguistics:
948 System Demonstrations. pp. 55–60.
- 949 Mesnil, G., He, X., Deng, L., Bengio, Y., 2013. Investigation of recurrent-neural-
950 network architectures and learning methods for spoken language understanding.
951 In: Proceedings of Interspeech. p. to appear.

- 952 Mikolov, T., Deoras, A., Kombrink, S., Burget, L., ernock, J., 2011a. Empiri-
953 cal evaluation and combination of advanced language modeling techniques. In:
954 Proceedings of Interspeech. pp. 605–608.
- 955 Mikolov, T., Deoras, A., Povey, D., Burget, L., Cernocký, J., 2011b. Strategies for
956 training large scale neural network language models. In: IEEE Workshop on
957 Automatic Speech Recognition and Understanding. pp. 196–201.
- 958 Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S., 2010. Recur-
959 rent neural network based language model. In: Proceedings of Interspeech. pp.
960 1045–1048.
- 961 Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S., 2011c. Exten-
962 sions of recurrent neural network language model. In: Proceedings of the IEEE
963 International Conference on Acoustics, Speech and Signal Processing. pp. 5528
964 –5531.
- 965 Mikolov, T., Zweig, G., 2012. Context dependent recurrent neural network lan-
966 guage model. In: IEEE Workshop on Spoken Language Technology. pp. 234–
967 239.
- 968 Mirowski, P., Chopra, S., Balakrishnan, S., Bangalore, S., 2010. Feature-rich con-
969 tinuous language models for speech recognition. In: Proceeding of IEEE Spo-
970 ken Language Technology Workshop. pp. 241–246.
- 971 Morin, F., Bengio, Y., 2005. Hierarchical probabilistic neural network language
972 model. In: AISTATS05. pp. 246–252.
- 973 Mousa, A. E.-D., Kuo, H.-K. J., Mangu, L., Soltau, H., 2013. Morpheme-based
974 feature-rich language models using deep neural networks for lvcsr of egyptian

- 975 arabic. In: Proceedings of IEEE International Conference on Acoustics, Speech
976 and Signal Processing. pp. 8435–8439.
- 977 Murphy, K. P., 2002. Dynamic bayesian networks: Representation, inference and
978 learning. Ph.D. thesis, University of California, Berkeley.
- 979 Ney, H., Essen, U., Kneser, R., 1994. On Structuring Probabilistic dependencies
980 in stochastic language modelling. *Computer Speech and Language* 8.
- 981 Niesler, T., Woodland, P. C., 1996. Combination of word-based and category-
982 based language models. In: Proceedings of International Conference on Spoken
983 Language. Vol. 1. pp. 220–223 vol.1.
- 984 Niesler, T. R., Whittaker, E. W. D., Woodland, P. C., 1998. Comparison of part-of-
985 speech and automatically derived category-based language models for speech
986 recognition. In: Proceedings of the IEEE International Conference on Acous-
987 tics, Speech and Signal Processing. pp. 177–180 vol.1.
- 988 Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J. P., Moortgat,
989 M., Baayen, H., 2002. Experiences from the Spoken Dutch Corpus project. In:
990 Araujo (eds), Proceedings of the Third International Conference on Language
991 Resources and Evaluation. pp. 340–347.
- 992 Pereira, F., Tishby, N., Lee, L., 1993. Distributional clustering of English words.
993 In: Association for Computational Linguistics. pp. 183–190.
- 994 Pietra, S. A. D., Pietra, V. J. D., Mercer, R. L., Roukos, S., 1992. Adaptive lan-
995 guage modeling using minimum discriminant estimation. In: Proceedings of
996 the workshop on Speech and Natural Language. pp. 103–106.

- 997 Putthividhya, D. P., Attias, H. T., Nagarajan, S., 2009. Independent factor topic
998 models. In: Proceedings of the 26th Annual International Conference on Ma-
999 chine Learning. ACM, pp. 833–840.
- 1000 Rosenfeld, R., 1996. A maximum entropy approach to adaptive statistical lan-
1001 guage modeling. *Computer, Speech and Language* 10(3), 187–228.
- 1002 Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. Learning representations
1003 by back-propagating errors. *Nature* 323, 533–536.
- 1004 Shi, Y., Larson, M., Jonker, C. M., 2014. Recurrent neural network language
1005 model adaptation with curriculum learning. *Computer Speech & Language* (0),
1006 —.
- 1007 Shi, Y., Wiggers, P., Jonker, C. M., 2010. Language modelling with dynamic
1008 bayesian networks using conversation types and part of speech information.
1009 In: *The 22nd Benelux Conference on Artificial Intelligence*. pp. 154–161.
- 1010 Shi, Y., Wiggers, P., Jonker, C. M., 2011. Combining topic specific language mod-
1011 els. In: *Proceedings of the International Conference on Text, Speech and Dia-*
1012 *logue*. pp. 99–106.
- 1013 Shi, Y., Wiggers, P., Jonker, C. M., 2012. Towards recurrent neural networks lan-
1014 guage models with linguistic and contextual features. In: *Proceedings of Inter-*
1015 *speech*. pp. 1664–1667.
- 1016 Shi, Y., Wiggers, P., Jonker, C. M., 2013. Classifying the socio-situational set-
1017 tings of transcripts of spoken discourses. *Speech Communication* 55 (10), 988
1018 – 1002.

- 1019 Shi, Y., Yao, K., Chen, H., Pan, Y.-C., Hwang, M.-Y., Peng, B., 2015. Contextual
1020 spoken language understanding using recurrent neural networks. In: Proceedings of the IEEE International Conference on Accoustic Speech and Signal
1021 processing. In: Proceedings of the IEEE International Conference on Accoustic Speech and Signal
1022 Processing.
- 1023 Sigurd, B., Eeg-Olofsson, M., Van Weijer, J., 2004. Word length, sentence length
1024 and frequency—Zipf revisited. *Studia Linguistica* 58 (1), 37–52.
- 1025 Steinbiss, V., Tran, B.-H., Ney, H., 1994. Improvements in beam search. In: Proceedings of the International Conference on Spoken Language Processing. pp.
1026 2143–2146.
1027 2143–2146.
- 1028 Su, Y., 2011. Knowledge Integration Into Language Models: A Random Forest
1029 Approach. *BiblioBazaar*.
- 1030 Udney Yule, G., 1939. On sentence-length as a statistical characteristic of style
1031 in prose: With application to two cases of disputed authorship. *Biometrika*
1032 30 (3/4), 363–390.
- 1033 Ueberla, J. P., 1995. More efficient clustering of n-grams for statistical language
1034 modeling. In: Proceedings of EUROSPEECH. pp. 1257–1260.
- 1035 van den Bosch, A., 2006. Scalable classification-based word prediction and con-
1036 fusible correction. *Traitement Automatique des Langues* 46 (2), 39–63.
- 1037 Van Eynde, F., 2004. Part of speech tagging en lemmatisering van het corpus
1038 gesproken nederlands. Tech. rep., K.U.Leuven.
- 1039 Wang, W., Harper, M. P., 2002. The superARV language model: Investigating the
1040 effectiveness of tightly integrating multiple knowledge sources. In: Proceedings

- 1041 of Conference of Empirical Methods in Natural Language Processing. pp. 238–
1042 247.
- 1043 Wang, W., Vergyri, D., 2006. The use of word n-grams and parts of speech for
1044 hierarchical cluster language modeling. In: Proceedings of IEEE International
1045 Conference on Acoustics, Speech and Signal Processing. Vol. 1. pp. I–I.
- 1046 Wiggers, P., Rothkrantz, L. J. M., 2006a. Dynamic bayesian networks for lan-
1047 guage modeling. In: Text and Speech and Dialogue. p. 555–562.
- 1048 Wiggers, P., Rothkrantz, L. J. M., 2006b. Topic-based language modeling with
1049 dynamic bayesian networks. In: Proceedings of the Ninth International Confer-
1050 ence on Spoken Language Processing. pp. 1866–1869.
- 1051 Wiggers, P., Rothkrantz, L. J. M., 2007. Exploratory analysis of word use and
1052 sentence length in the spoken Dutch corpus. In: Proceedings of the International
1053 Conference on Text, Speech and Dialogue. pp. 366–373.
- 1054 Wu, Y., Lu, X., Yamamoto, H., Matsuda, S., Hori, C., Kashioka, H., 2012. Fac-
1055 tored language model based on recurrent neural network. In: Proceedings of
1056 International Conference of Computational Linguistics. pp. 2835–2850.
- 1057 Xu, P., Jelinek, F., 2004. Random forests in language modeling. In: Proceedings
1058 of EMNLP. pp. 325–332.
- 1059 Xu, P., Karakos, D., Khudanpur, S., 2009. Self-supervised discriminative train-
1060 ing of statistical language models. In: IEEE Workshop on Automatic Speech
1061 Recognition and Understanding. pp. 317–322.

- 1062 Yamamoto, H., Sagisaka., Y., 1999. Multi-class composite n-gram based on con-
1063 nection direction. In: Proceedings of 1999 IEEE International Conference on
1064 Acoustics, Speech, and Signal Processing. pp. 533–536.
- 1065 Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., Yu, D., 2013. Recurrent neural net-
1066 works for language understanding. In: Proceedings of Interspeech. pp. 2524–
1067 2528.
- 1068 Zipf, G. K., 1949. Human Behavior and the Principle of Least Effort. Addison-
1069 Wesley (Reading MA).