Computational Linguistics in the Netherlands Journal 4 (2014) 91-104 Submitted 06/2014; Published 12/2014

The effect of word similarity on N-gram language models in Northern and Southern Dutch

Joris Pelemans* JORIS.PELEMANS@ESAT.KULEUVEN.BE
Kris Demuynck** KRIS.DEMUYNCK@ELIS.UGENT.BE
Hugo Van hamme* HUGO.VANHAMME@ESAT.KULEUVEN.BE
Patrick Wambacq* PATRICK.WAMBACQ@ESAT.KULEUVEN.BE

* *Department of ESAT, KU Leuven, Leuven, Belgium*

** *ELIS, Ghent University, Ghent, Belgium*

Abstract

In this paper we examine several combinations of classical N-gram language models with more advanced and well known techniques based on word similarity such as cache models and Latent Semantic Analysis. We compare the efficiency of these combined models to a model that combines N-grams with the recently proposed, state-of-the-art neural network-based continuous skip-gram. We discuss the strengths and weaknesses of each of these models, based on their predictive power of the Dutch language and find that a linear interpolation of a 3-gram, a cache model and a continuous skip-gram is capable of reducing perplexity by up to 18.63%, compared to a 3-gram baseline. This is three times the reduction achieved with a 5-gram.

In addition, we investigate whether and in what way the effect of Southern Dutch training material on these combined models differs when evaluated on Northern and Southern Dutch material. Experiments on Dutch newspaper and magazine material suggest that N-grams are mostly influenced by the register and not so much by the language (variety) of the training material. Word similarity models on the other hand seem to perform best when they are trained on material in the same language (variety).

1. Introduction

A statistical language model (LM) is a model that assigns a probability to a sequence of words. This is a task that is useful in many areas of research including spelling correction, optical character recognition, handwriting recognition, machine translation and speech recognition. The assigned probability allows the prediction of the next word in a sentence which may be used to steer pattern recognition applications towards a word hypothesis that is more linguistically plausible. Although humans are capable of capturing detailed information from a sentence by dissecting the individual words and building them into complex, meaningful concepts that can be mapped to known concepts in the world, machines have yet to learn such sophisticated relationships and are hitherto doomed to find salvage in simpler techniques.

One such technique that has proven surprisingly fruitful is the N-gram language model. This model makes the naive assumption that natural language consists of overlapping sequences of N words or N-grams and that to predict the next word, one need only look at the previous $N-1$ words instead of the whole history. Although this approximation is obviously a crude one, N-gram LMs are easy to create and can be readily applied to many complex tasks with remarkable success. For this reason, they have been the workhorse in the field of automatic speech recognition for the last couple of decades.

Notwithstanding their benefits, N-gram LMs suffer from several deficiencies, one of which is data sparsity. N-gram LM probabilities are estimated from their relative occurrences in a large corpus and since language is such an unconstrained and creative process, there is simply not enough

data available to reliably estimate the probabilities for all N-grams. Even with the Big Data that is available nowadays via the Internet, many of the possible N-grams of a language do not occur (enough), no matter how small N is chosen (Allison et al. 2006). Many methods have been suggested to alleviate this problem e.g. smoothing (Witten and Bell 1991, Good 1953, Kneser and Ney 1995, Chen and Goodman 1999), word clustering (Brown et al. 1992, Pelemans et al. 2014), skip N-grams (Huang et al. 1992, Rosenfeld 1994, Ney et al. 1994), and more recently continuous models (Bengio et al. 2003, Mikolov et al. 2010), but the issue of language data sparsity is still very much an open one.

Another flaw with N-gram LMs is their primary assumption that the history upon which the prediction is based, can be reduced to only a handful of words. Although it may be the case that much if not most of the information (Rosenfeld 1994) resides in the immediate, local context of the current word, other long-span phenomena such as sentence- or document-level semantic relations can only be modeled with the help of the more distant history. One of the ways to address this problem, is to combine N-gram LMs with techniques that model word similarity in the hope of capturing both local and global phenomena.

The focus of this paper is to examine the combination of N-gram LMs with several of these techniques including well established models such as cache models (Kuhn and de Mori 1990) and Latent Semantic Analysis (Deerwester et al. 1990). The combined models are compared to a new model that combines N-grams with the recently proposed continuous skip-gram model (Mikolov et al. 2013a, Mikolov et al. 2013b, Mikolov et al. 2013c), which is a scalable adaptation of a Neural Network Language Model (NNLM) (Bengio et al. 2003) and the current state-of-the-art in word similarity. We discuss the strengths and weaknesses of each of these combined models, based on their predictive power of the Dutch language and investigate whether and in what way the effect of Southern Dutch training material differs when evaluated on Northern and Southern Dutch material.

The remainder of this paper is organized as follows. In Section 2 we give an overview of all of the investigated word similarity models, except for the continuous skip-gram model which we explain in more detail in Section 3. Section 4 covers in depth analysis of each of the models and compares their predictive power of the Dutch language. We end with a conclusion and some future work.

2. Classical word similarity models

In this section we discuss several classical word similarity models that are less accurate, but also computationally less expensive than NNLMs and that have been shown to complement well with N-gram language models. Because these models can be integrated directly into a speech decoder, they have access to the complete search space and may therefore achieve competitive results in a single-pass scenario. Alternatively, they can also be used as a first pass LM, after which an NNLM may still be used in a second, rescoring pass.

2.1 Cache models

Cache models are based on the observation that topical words tend to re-occur within a text. A cache memory is kept that keeps track of the last K words in a document and is consulted when predicting the next word. In their simplest form (Kuhn and de Mori 1990), cache models distribute the probability mass uniformly among all the K tokens in the cache memory:

$$P_{cache}(w_q|w_{q-K}^{q-1}) = \frac{C_{cache}(w_q)}{K} \tag{1}$$

where $C_{cache}(w_q)$ indicates the frequency of word w_q in the cache memory.

Extensions exist (Clarkson and Robinson 1997) where word age is taken into account i.e. the impact of older tokens is decreased by applying a (typically exponential) decay weighting function:

$$P_{cache}(w_q|w_{q-K}^{q-1}) = \beta \sum_{j=q-K}^{q-1} I(w_q = w_j)\alpha^{(q-j)} \quad (2)$$

where $I(x)$ is an indicator function that returns 1 if x is true and 0 otherwise, α is the decay rate and β is a normalization factor.

Cache models are simple and efficient and are capable of modeling long distance phenomena which is one of the main weaknesses of N-gram LMs. They do not however employ any kind of semantic knowledge.

2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Deerwester et al. 1990) is one of the earliest attempts to discover hidden semantic structure in a text by considering word co-occurrences. It is a dimensionality reduction technique based on truncated Singular Value Decomposition that is applied on a term-document matrix W which contains in each cell the number of times a word (rows) occurs in a document (columns). As is shown in Figure 1, this frequency matrix is decomposed into a product of an orthonogal matrix U, a diagonal matrix S, containing the singular values, and another orthogonal matrix V^T. The information in W that corresponds to the smallest singular values is considered to be noise induced by data errors and word redundancy and is effectively removed by preserving only the k largest singular values. The resulting rank k approximation is optimal w.r.t. the Frobenius norm and uncovers latent semantic relations between words and documents.

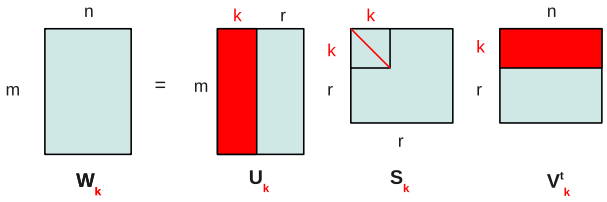

Figure 1: (Truncated) Singular Value Decomposition

Often a preprocessing step is useful, because the documents are not of equal length and not all words are equally informative. To this end, the raw counts of W may be transformed according to a weighting scheme which typically consists of a global component $G(i)$ and a local component $L(i, j)$ (Dumais 1991):

$$W'(i, j) = G(i)L(i, j) \quad (3)$$

where W' represents the frequency matrix W after applying the weighting scheme. The same global weight – indicating the overall importance of a term – is applied to an entire row of the matrix, whereas the local weight – which indicates the importance of a term in a specific document – is applied to each cell in the matrix.

Although many different schemes exist, the choice is not so critical to the overall performance of LSA and in this paper we only investigate the use of TF-IDF, which is one of the most used weighting schemes in the context of information retrieval. For an overview and classification of different weighting schemes, we refer to (Salton 1971).

Semantic similarity between documents and words is measured by calculating the cosine distance in the latent space. In the context of language modeling, the history is considered to be a (pseudo-)document $\tilde{d}_{q-1}$ and the cosine distance K between the word vector u_q and the (pseudo-)document vector $\tilde{v}_{q-1}$ in the latent space is converted into a probability as follows (Coccaro and Jurafsky 1998):

$$P(w_q|\tilde{d}_{q-1}) = \frac{[K(u_q, \tilde{v}_{q-1}) - \min_u K(u, \tilde{v}_{q-1})]^\gamma}{\sum_{w_i \in \mathcal{V}} [K(u_i, \tilde{v}_{q-1}) - \min_u K(u, \tilde{v}_{q-1})]^\gamma} \quad (4)$$

where $\mathcal{V}$ is the vocabulary and γ is a parameter that controls the dynamic range of the distribution.

Although LSA is capable of uncovering semantic relations between words and documents, it is insensitive to the multiple senses that many words have. Moreover, using the Frobenius norm as an error function assumes normally distributed data with independent entries, which is not the case for the counts in the term-document matrix.

3. Continuous skip-gram model

Motivated by the recent successes in neural network language modeling, Mikolov et al. (Mikolov et al. 2013a, Mikolov et al. 2013b, Mikolov et al. 2013c) recently proposed a new architecture for the acquisition of high-quality word embedding vectors which might not be able to represent the data as precisely as neural networks, but is less complex and can therefore handle more data efficiently. The continuous skip-gram model (CSM) is the current state-of-the-art in word similarity and is shown in Figure 2. It is a log-linear classifier with a continuous projection layer that tries to maximize the prediction of words within a range R before and after the current word. Similar to the Recurrent NNLM (Mikolov et al. 2010, Mikolov et al. 2011), the network is trained by using backpropagation with stochastic gradient ascent until convergence and uses the softmax activation function to ensure that the output layer forms a valid probability distribution:

$$p(w_{t+j}|w_t) = \frac{\exp(v_{w_t}^T v'_{w_{t+j}})}{\sum_{w=1}^{W} \exp(v_{w_t}^T v'_{w_{t+j}})} \quad (5)$$

where v_w and v'_w are the input and output representations of word w.

The model was made more efficient by approximating the full softmax by a hierarchical version which was first introduced by Morin and Bengio (Morin and Bengio 2005). The hierarchical softmax uses a binary Huffman tree representation of the output layer with the words as its leaves and for each node represents the relative probabilities of its child nodes explicitly. This has a significant positive effect on the overall speed of the model. For more information on the hierarchical softmax and the continuous skip-gram model in general, we refer the reader to (Mikolov et al. 2013b) and (Mikolov et al. 2013a).

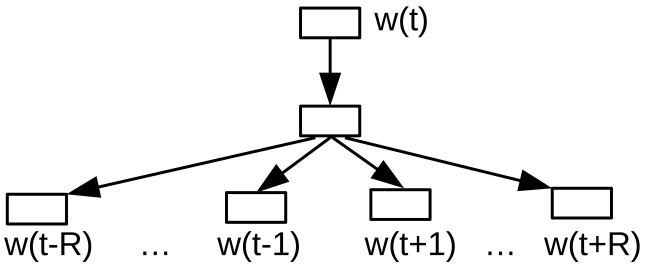

Figure 2: Continuous Skip-gram Model

An interesting observation was made in (Mikolov et al. 2013c): the vector-space word representations that are implicitly learned by the input-layer weights are surprisingly good at capturing

both semantic and syntactic regularities in language and each relationship is characterized by a relation-specific vector offset. This allows intuitive vector mathematics based on the offsets between words e.g. *King - Man + Woman* results in a vector that is very close to that of *Queen*. It is clear that these CSM word embedding vectors can be a valuable source of information to enrich existing language modeling techniques.

For word embedding vectors to be incorporated into a language model, we need to have a representation at a higher level than just individual words. A naive notion of a document can be achieved by calculating the centroid of the previous K word vectors, but this makes the unreasonable assumptions that the meaning of a phrase is equal to the sum of its components and that all words in the history are equally important.

In (Mikolov et al. 2013a), the authors show that the first assumption can be overcome in part by detecting common phrases i.e. words that appear frequently together and infrequently in other contexts. This way they were able to replace *New York Times* by a single token while *this is* remained unchanged.

They did not however address the assumption of equal word importance. Function words like *the* and *of* are clearly less informative than content words, hence should be given less weight. This can be dealt with quite easily by applying the same TF-IDF weighting as was mentioned in Section 2.2.

Finally, one may wonder whether the vector addition of many words is still meaningful. It is likely that the context used in language models based on word embedding vectors should either be limited in length or processed in some hierarchical way where documents are recursively decomposed into smaller meaningful units.

4. Experimental validation

In this work we compare the combination of the beforementioned word similarity techniques with N-gram language models. Model fitness is measured as test set perplexity and compared with a 3-gram baseline model with modified Kneser-Ney smoothing. This form of smoothing has been shown to outperform the other well-known smoothing methods on a number of occasions and our case did not prove any different. We also trained 4-gram and 5-gram language models, but since the reduction in perplexity was minimal and since we are mostly interested in the reduction caused by adding word similarity information, we chose the 3-gram model to interpolate with the other models. We do however compare our final interpolated models to a 4-gram and 5-gram LM to show that our techniques can be more interesting than simply increasing the LM context.

All of the trained models are combined using linear interpolation. Although non-linear interpolation has been shown to achieve larger perplexity reductions (Coccaro and Jurafsky 1998), this better performance comes at the cost of higher complexity, since the LM scores need to be normalized for each evaluation. As our final goal is to apply these models in the context of automatic speech recognition, we prefer a fast combination technique. Note that whenever we report interpolation weights, we mean the weight attributed to the model currently under discussion and not the weight of the N-gram LM.

Our LM training data consists of a collection of normalized newspaper texts from the Flemish digital press database Mediargus. All of the models were trained on excerpts from the Southern Dutch newspaper De Standaard, which contain 65M word tokens. Vocabulary selection was based on the most frequent words in this data set. Parameter optimization was done on a development set consisting of excerpts from the Southern Dutch newspaper De Morgen, which contain 100k word tokens. Finally, we validated all of our models on excerpts of the Southern Dutch magazine Knack and Northern Dutch newspaper NRC, both of which were limited to 50k word tokens. In addition, we double-checked our conclusions by looking at similar length excerpts of two Southern Dutch newspapers: Gazet van Antwerpen and Het Belang van Limburg. None of the data sets were marked with document boundaries, but instead a document length of 30 sentences was assumed.

	α							
K	0.95	0.97	0.99	0.992	0.994	0.996	0.998	1
0 (3-gram)	222.02	222.02	222.02	222.02	222.02	222.02	222.02	222.02
50	214.49	213.48	212.94	212.92	212.90	212.89	212.88	212.88
250	208.70	204.26	200.46	200.39	200.44	200.61	200.91	201.36
500	208.70	204.15	198.85	198.59	198.57	198.91	199.72	201.04
1000	208.70	204.15	198.66	198.26	198.08	198.43	199.90	203.19
2000	208.70	204.15	198.66	198.25	198.05	198.35	200.17	206.81

Table 1: Perplexity results for interpolated N-gram+cache models with a vocabulary of 100000 words, as a function of decay rate α and cache memory size K, as measured on the De Morgen development set.

All of the word similarity models were trained using the open-source Python framework gensim (Řehůřek and Sojka 2010), which contains intuitive and scalable implementations for many popular NLP models. The N-gram language models were trained with the SRILM toolkit (Stolcke 2002).

The models using LSA and CSM are all preprocessed using TF-IDF. No phrase detection whatsoever was performed.

For each of the combined models, we performed an exhaustive grid search in their parameter space for vocabularies of size 50000, 100000 and 200000. Unless mentioned otherwise, we only show the results of a 100000 words vocabulary for the sake of clarity. We investigate what the optimal values are, which parameters are most influential and whether or not these parameters are robust w.r.t. vocabulary size and perplexity. We also compare common parameters across models e.g. dimensionality and interpolation weight. Note that whenever a particular parameter is discussed, the remaining parameters are set to their optimal value w.r.t. to the first parameter, thus avoiding any opaque results by averaging effects.

The experiments can be summarized as follows: first we explore cache models and investigate whether exponential decay is worth the extra computation. In Section 4.2 we compare the dimensionality of LSA and CSM. Next, in Section 4.3, we study the effect of the context window. Then, in Section 4.4 we compare each of the models w.r.t. to their complexity. Finally, we review the best models, discuss their performance on two test sets and make some general conclusions on the applicability of Southern Dutch training data for Northern Dutch applications.

4.1 Cache models: to forget or not to forget

Cache models have been shown to work better when attributing less weight to older words i.e. using a word influence decay (Clarkson and Robinson 1997). In this section, we first investigate to what extent this is true for our data and which type of decay is the most promising. We also examine some general properties of our cache models.

Table 1 shows the perplexity of different interpolated N-gram+cache models on the development set as a function of the decay rate α and the size of the cache memory K. No decay is indicated by a value of $\alpha = 1$, all the other values represent exponential decay. As can be seen, the difference in performance between no decay and exponential decay is negligible. No decay performs better on somewhat smaller memories; most likely they attribute too much weight to older, less relevant words. Exponential decay is capable of capturing more distant information with a minimal perplexity for $\alpha = 0.994$ and $K = 2000$, although the gain by increasing the size of the memory is minimal. Notice that the perplexity results for lower values of α are identical for large memories. This is because the decay is so rapid that the influence of older words quickly reaches zero.

The optimal parameters of the cache models are more or less constant over different vocabulary sizes and are relatively robust with a maximal perplexity difference of around 20. The worst per-

	dimensionality					
	50	150	250	500	750	1000
3-gram	222.02	222.02	222.02	222.02	222.02	222.02
3-gram+LSA	210.93	205.43	203.35	201.64	201.19	200.85
3-gram+CSM	209.28	204.29	201.51	197.58	196.54	195.38

Table 2: Perplexity results for N-gram LMs, interpolated with LSA and CSM with a vocabulary of 100000 words, as a function of dimensionality, as measured on the De Morgen development set.

formance is observed for setups with a high cache interpolation weight and a very short window or low α. The optimal cache interpolation weight is 0.1 for most setups. The best setup is capable of reducing the perplexity by 24 over the 3-gram LM baseline.

We conclude that when choosing a setup for cache models, exponential decay is only marginally better than no decay. However, since the added complexity is minimal, the only reason not to use it is when parameter optimization is not an option. In the following experiments, we will therefore only refer to the optimal cache model with exponential decay.

4.2 On the dimensionality of word similarity models

In this section, we investigate how many features LSA and CSM need to store useful information about words. Less features needed means less storage needed which is a desirable property, but do more features always contain more information? What is the optimal dimensionality for each model, how do they compare to each other and which model has the highest information-per-feature ratio i.e. which model is able to get the best results with the least amount of features?

Table 2 shows the results of different models using either LSA or CSM as word similarity component. Their performance is measured as perplexity on the development set as a function of the dimensionality. It can be seen that the models using LSA already perform well at a dimensionality of 50, but very quickly reach saturation around 150 with a marginal perplexity reduction for a higher number of dimensions. The word embedding vectors also perform well at a dimensionality of 50, but they gain more by additional dimensions until a dimensionality of around 500. This causes them to outperform LSA by an increasing margin, albeit small, even at the highest number of dimensions.

The optimal parameters of the different models are, as was the case with the cache models, similar over different vocabulary sizes with, for LSA, a window of 200, LSA interpolation weight of 0.08, dimensionality of 1000 and $\gamma = 7$, and, for CSM, a window of 100, CSM interpolation weight of 0.07, dimensionality of 1000 and training context R of 240. The robustness of the CSM parameters is comparable to that of cache models with a maximal perplexity difference of around 25. The parameters of LSA are even more robust than those of cache models with a maximal perplexity difference of only around 10.

We conclude that CSM is capable of finding more interesting features in the data, yielding a small advantage over LSA in our development set. We believe that this difference will manifest itself more clearly as the size of the data increases.

4.3 Windows: how much we can learn from the distant past

Each of the trained models uses a certain amount of the most recently uttered words to evaluate which word is most likely to be uttered next i.e. the context window. N-gram LMs are forced to keep the size of their window very small, because they preserve the word order within the window which rapidly leads to sparsity problems in higher order windows. Word similarity models are typically bag-of-words models i.e. they do not preserve the word order within the window and therefore do

	window size							
	50	100	150	200	250	500	1000	2000
3-gram	222.02	222.02	222.02	222.02	222.02	222.02	222.02	222.02
3-gram+cache	212.88	205.73	202.81	201.43	200.44	198.57	198.08	198.05
3-gram+LSA	204.98	201.73	201.10	200.85	201.83	203.66	206.77	210.58
3-gram+CSM	196.28	195.38	195.49	195.95	196.77	200.98	205.73	211.17

Table 3: Perplexity results for N-gram LMs, interpolated with cache models, LSA and CSM with a vocabulary of 100000 words, as a function of window size, as measured on the De Morgen development set.

not suffer from this restriction. It is not clear however how much information resides in the very distant context words and whether a model is able to extract this information. In this section, we investigate what the optimal window sizes are for all three of the examined models and attempt to explain any differences therein. We discuss why certain models are not capable of extracting valuable information from distant context as well as others.

Table 3 shows the perplexity results of all the models on the development set, as a function of the size of the window. As was already discovered in Section 4.1, cache models with exponential decay are capable of retrieving valuable information from increasingly large contexts. The inverse is true for LSA and CSM, where we observe a rapid saturation and even degradation with increasing window size. We assume this is because, as opposed to cache models, these models capture real semantic information which is related to the ongoing topic. In natural language, and in particular in newspaper material such as our development set, these topics can change quite quickly. We believe this is also one of the reasons why cache models often outperform real similarity models, as they are able to attribute high probabilities to function words as well as content words.

LSA and CSM behave quite similarly w.r.t. context window size. Both saturate around 150 and then slowly degrade. Unexpectedly, CSM does not degrade more rapidly than LSA. As we mentioned in Section 3, we expected that averaging vectors for an increasing amount of words would yield a word vector which could be described as having all meanings and no meaning at all. Our results show that this is not the case and, even though there is no document concept in these models, degradation behavior is quite similar to LSA. We believe that this is largely due to the application of TF-IDF preprocessing, which reduces the effect of most of the words in the window.

Finally, we observe that the largest perplexity reduction is achieved by CSM, albeit by a small margin. Given the high quality of the vectors, we believe that smarter compositional models for word embedding vectors will lead to a more significant difference in performance.

We conclude that unlike cache models, TF-IDF preprocessed LSA and CSM thrive in smaller contexts of several hundreds of words and that CSM is capable of making the most out of the least number of words.

4.4 Complexity

In this section, we sidetrack for a moment from perplexity evaluation and investigate a more practical concern which is model complexity. We do not attempt to theoretically analyze the asymptotic computational complexity of each of the models, but rather we compare the processing time needed for each of the models to evaluate the Knack test set. Each reported processing time is the result of averaging the times of 10 otherwise identical experiments.

Table 4 shows the processing time for different models, as measured on an Intel Core i5-2400 processor with 1 core only. We compared the times for N-gram models interpolated with a cache model with a window of 2000 words to LSA and CSM with a window of 150 words and 500 dimensions.

	processing time
3-gram+cache	1m50
3-gram+LSA	2h06m
3-gram+CSM	1h03m

Table 4: Processing times for N-gram LMs, interpolated with cache models, LSA and CSM with a vocabulary of 100000 words, as measured on the Knack test set, using a Intel Core i5-2400 processor with 1 core.

The simplicity of the cache model is apparent as it needs less than 2 minutes for a complete evaluation of the test set. LSA takes on average 2h06 to complete the evaluation, whereas CSM is capable of finishing the job in 1h03, exactly half the processing time of LSA.

It is clear that CSM is nowhere near as efficient as cache models, but it is twice as fast as LSA, which makes it an interesting candidate for integration into a speech recognizer.

4.5 Test set evaluation

In this section we evaluate all of the created models on two different test sets: the Southern Dutch magazine Knack and the Northern Dutch newspaper NRC. We first compare the interpolated models to the 3-gram LM baseline to highlight the reduction in perplexity. We also report perplexities for 4-gram and 5-gram LMs (both with modified Kneser-Ney smoothing) to indicate that our techniques cannot be simply matched by increasing the context of an N-gram LM. Next, we show that the best two models (cache and CSM) are complementary and that a combined model yields the best results. All of the experiments are repeated for vocabularies of 50000, 100000 and 200000 to investigate any effects due to vocabulary size. Finally, we compare the results for both test sets in an attempt to find out to what extent models built with Southern Dutch material are suitable for Northern Dutch and double-check our conclusions on two additional Southern Dutch newspapers.

	Knack		NRC	
	PPL	Gain	PPL	Gain
3-gram	210.36		201.75	
4-gram	198.10	5.83%	189.20	6.22%
5-gram	197.69	6.02%	188.61	6.51%
3-gram+cache	187.89	10.68%	180.59	**10.49%**
3-gram+LSA	186.12	11.53%	185.26	8.17%
3-gram+CSM	183.62	**12.71%**	181.84	9.87%
3-gram+cache+CSM	175.82	16.42%	171.52	14.98%
4-gram+cache+CSM	166.20	21.00%	161.79	19.81%
5-gram+cache+CSM	166.16	21.02%	161.66	19.87%

Table 5: Perplexity results for all of the models with a vocabulary of 50000 words, as measured on the Knack and NRC test sets.

Tables 5, 6 and 7 show perplexity results for all of the mentioned models on both test sets, for vocabulary sizes of 50000, 100000 and 200000 words respectively. It can be seen that all of the interpolated models outperform the 3-gram LM baseline as well as 4-gram and 5-gram LMs. This is true regardless of the vocabulary size and shows that the reported perplexity reductions of 7.63% to 13.32% cannot be achieved by simply increasing the N-gram context. Zooming in on the interpolated models, we notice that LSA consistently scores lowest, except for Knack with a vocabulary of 50000 words, where it does better than the cache model. CSM is always the best

	Knack		NRC	
	PPL	Gain	PPL	Gain
3-gram	241.83		233.05	
4-gram	227.62	5.88%	218.60	6.20%
5-gram	227.05	6.11%	217.79	6.55%
3-gram+cache	213.27	11.81%	205.39	**11.87%**
3-gram+LSA	214.39	11.35%	213.35	8.45%
3-gram+CSM	209.68	**13.29%**	208.24	10.64%
3-gram+cache+CSM	199.05	17.69%	194.60	16.50%
4-gram+cache+CSM	188.08	22.23%	183.61	21.21%
5-gram+cache+CSM	187.96	22.28%	183.37	21.32%

Table 6: Perplexity results for all of the models with a vocabulary of 100000 words, as measured on the Knack and NRC test sets.

	Knack		NRC	
	PPL	Gain	PPL	Gain
3-gram	267.00		261.37	
4-gram	251.15	5.93%	245.16	6.20%
5-gram	250.43	6.20%	244.23	6.56%
3-gram+cache	231.96	13.12%	226.55	**13.32%**
3-gram+LSA	237.50	11.05%	241.43	7.63%
3-gram+CSM	231.79	**13.19%**	235.14	10.04%
3-gram+cache+CSM	217.24	18.63%	216.00	17.36%
4-gram+cache+CSM	205.16	23.16%	203.77	22.04%
5-gram+cache+CSM	204.97	23.23%	203.49	22.14%

Table 7: Perplexity results for all of the models with a vocabulary of 200000 words, as measured on the Knack and NRC test sets.

model for Knack, although only by a small margin when using a vocabulary of 200000 words. Cache models have no competition whatsoever when evaluated on NRC. Finally, it is clear that cache models and CSM are complementary as the models that combine both cache and CSM outperform the others by a large degree with perplexity reductions of up to 23%.

The size of the vocabulary has the largest effect on the cache models, with performance going up with increasing vocabulary size. This is a logical result as more words can be stored in the cache memory and it is also visible in the models combining both cache and CSM. LSA and CSM do not seem to be affected much by vocabulary size, although there is some evidence that performance drops somewhat when using too many words.

When we compare the results of Knack with those of NRC, it can be seen from any of the three tables, that the 3-gram LM baseline yields the lowest perplexity on the Northern Dutch newspaper NRC. As the training set also consisted of newspaper excerpts, this leads us to believe that N-gram LMs capture language information that is mostly related to the register of the training material. We assume that both the word choice and the syntactic structure of a newspaper are somewhat more formal than those of a magazine. The fact that Southern and Northern Dutch differ in word usage does not have any observable effect, perhaps due to the nature of the material. It is likely that newspapers, with their formal registers and broad target group, tend to choose more standard words, thus diminishing this language discrepancy.

	Gazet van Antwerpen		Het Belang van Limburg	
	PPL	Gain	PPL	Gain
3-gram	205.31		199.99	
4-gram	192.79	6.09%	185.08	7.46%
5-gram	192.47	6.25%	184.62	7.69%
3-gram+cache	180.92	11.88%	180.16	9.92%
3-gram+LSA	183.32	10.71%	181.63	9.18%
3-gram+CSM	180.72	11.98%	177.42	11.29%

Table 8: Perplexity results for all of the models with a vocabulary of 50000 words, as measured on the additional Southern Dutch news papers.

If we look at the effect of the different models on the baseline, we notice that cache models have a similar effect on both test sets, which is not surprising as cache models are not trained and words are likely to be repeated in most if not all languages. We do however observe a large discrepancy in the word similarity models where both LSA and CSM show a larger perplexity reduction on the Southern Dutch magazine Knack. This is a clear indication that there is in fact a difference between Northern and Southern Dutch w.r.t. language modeling. As both these models are based on the idea of co-occurrence, this seems to indicate that, for our data, words co-occur differently in Southern Dutch than they do in Northern Dutch. It is possible that this is due to a topic effect i.e. that certain topics occur more often in Southern Dutch texts, for example because they address a regional phenomenon. Another explanation is that, although there does not seem to be a large discrepancy in the used vocabulary between the Southern Dutch training set and the Northern Dutch test set, there might be differences in meaning which yields different lexical relations.

Whatever the underlying cause, we conclude that Southern Dutch data can be used successfully as training material for the evaluation of Northern Dutch, especially when the intended models are N-grams and when there is a similarity in register. When the intended models concern word similarity, Southern Dutch texts can still prove valuable, but Northern Dutch data is more likely to be preferred.

To give further confidence to these conclusions we tested our models on two additional Southern Dutch newspapers: Gazet van Antwerpen and Het Belang van Limburg. Table 8 shows the perplexity results for these two test sets for a vocabulary size of 50000 words. It can be seen that the 3-gram LM baseline yields perplexities that are comparable to that of NRC. This strengthens our claim that N-gram LMs are hardly if at all influenced by language variety, but are in fact affected by style. On the other hand, when we look at the word similarity models, the perplexity reductions strongly resemble those of Knack, especially for CSM, reflecting the fact that they are all Southern Dutch publications.

5. Conclusions and future work

We examined several combinations of classical N-gram language models with word similarity models including cache models, Latent Semantic Analysis (LSA) and the recent continuous skip-gram model (CSM), and made several conclusions.

First, we did a thorough investigation of the model parameters and found that (1) cache models perform best when using exponential decay, although only marginally better than without decay; (2) CSM is making optimal use of larger dimensions which yields it a small, but consistent perplexity reduction over LSA; (3) both CSM and LSA are optimal with low context windows whereas cache models typically prefer longer windows.

Next, we compared the addition of each model to a 3-gram LM with modified Kneser Ney and found that the continous skip-gram model is a viable alternative for LSA in the context of interpolated language models. Not only does it achieve consistenly lower perplexities, it is also twice as fast and combines well with cache models. Our final model which uses an interpolation of a 3-gram LM, a cache model and a continuous skip-gram model was able to reduce the perplexity of a 3-gram LM by as much as 18.63%. This is about three times the reduction achieved with a 5-gram LM.

Finally, we compared the effect of LMs trained with Southern Dutch material on both a Southern Dutch and a Northern Dutch test set. We conclude that there is no outspoken difference in performance for N-gram LMs, which suggests that they are mostly influenced by the register of the training material. When it comes to word similarity on the other hand, material of the same language (variety) seems to be desirable. These conclusions were further confirmed by looking at two additional Southern Dutch newspapers.

Although in this paper, we did a thorough study of cache models, LSA and CSM and their properties, a lot of aspects still remain unexplored. As we mentioned, document boundaries were not marked in our training data, but instead a document was assumed to consist of 30 sentences. It would be interesting to see what the effect is of actual document boundaries or various, analogous assumptions. Moreover, we only employed decay on cache models, but this could certainly also be tested on LSA and CSM. We also want to investigate whether an increasing amount of data yields a proportional reduction in perplexity. Finally, and perhaps most importantly, we want to focus on compositionality, in an attempt to do smart addition of word vectors, which we believe is the next step in models using word embedding vectors. In future work, many if not all of these aspects will be explored further.

References

Allison, Ben, David Guthrie, and Louise Guthrie (2006), Another look at the data sparsity problem., *Proc. TSD*, Vol. 4188 of *Lecture Notes in Computer Science*, Springer, pp. 327–334.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin (2003), A neural probabilistic language model, *Journal of Machine Learning Research* **3**, pp. 1137–1155.

Brown, Peter F., Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai (1992), Class-based n-gram models of natural language, *Computational Linguistics* **18**, pp. 467–479.

Chen, Stanley F. and Joshua Goodman (1999), An empirical study of smoothing techniques for language modeling, *Computer Speech & Language* **13** (4), pp. 359–393.

Clarkson, P.R. and A. J. Robinson (1997), Language model adaptation using mixtures and an exponentially decaying cache, *Proc. ICASSP*, pp. 799–802.

Coccaro, Noah and Daniel Jurafsky (1998), Towards better integration of semantic predictors in statistical language modeling, *Proc. ICSLP*.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman (1990), Indexing by latent semantic analysis, *Journal of the American Society for Information Science* **41** (6), pp. 391–407.

Dumais, Susan T. (1991), Improving the retrieval of information from external sources, *Behavior Research Methods, Instruments, & Computers* (2), pp. 229–236, Psychonomic Society.

Good, I.J. (1953), The population frequencies of species and the estimation of population parameters, *Biometrika* **40**, pp. 237–264.

Huang, Xuedong, Fileno Alleva, Hsiao wuen Hon, Mei yuh Hwang, and Ronald Rosenfeld (1992), The sphinx-ii speech recognition system: An overview, *Computer, Speech and Language* **7**, pp. 137–148.

Kneser, Reinhard and Hermann Ney (1995), Improved backing-off for m-gram language modeling, *Proc. ICASSP*, Vol. I, pp. 181–184.

Kuhn, Roland and Renato de Mori (1990), A cache-based natural language model for speech recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **12** (6), pp. 570–583.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013a), Distributed representations of words and phrases and their compositionality, *Proc. NIPS*, pp. 3111–3119.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b), Efficient estimation of word representations in vector space, *CoRR*.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur (2010), Recurrent neural network based language model, *Proc. Interspeech*, pp. 1045–1048.

Mikolov, Tomas, Stefan Kombrink, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur (2011), Extensions of recurrent neural network language model, *Proc. ICASSP*, pp. 5528–5531.

Mikolov, Tomas, Wen tau Yih, and Geoffrey Zweig (2013c), Linguistic regularities in continuous space word representations, *Proc. HLT-NAACL*, pp. 746–751.

Morin, Frederic and Yoshua Bengio (2005), Hierarchical probabilistic neural network language model, *Proc. AISTATS*, pp. 246–252.

Ney, Hermann, Ute Essen, and Reinhard Kneser (1994), On structuring probabilistic dependences in stochastic language modelling., *Computer Speech & Language* **8** (1), pp. 1–38.

Pelemans, Joris, Kris Demuynck, Hugo Van hamme, and Patrick Wambacq (2014), Coping with language data sparsity: Semantic head mapping of compound words, *Proc. ICASSP*.

Řehůřek, Radim and Petr Sojka (2010), Software framework for topic modelling with large corpora, *Proc. LREC*, pp. 45–50.

Rosenfeld, R. (1994), *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, PhD thesis, Carnegie Mellon University.

Salton, G. (1971), *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Stolcke, Andreas (2002), SRILM - An Extensible Language Modeling Toolkit, *Proc. ICSLP*, pp. 257–286.

Witten, I. H. and T. C. Bell (1991), The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression, *IEEE Transactions on Information Theory* **37** (4), pp. 1085–1094.