The Seemingly (Un)systematic Linking Element in Danish

Sidsel Boldsen and **Manex Agirrezabal**
Centre for Language Technology
University of Copenhagen
{sbol,manex.aguirrezabal}@hum.ku.dk

Abstract

The use of a linking element between compound members is a common phenomenon in Germanic languages. Still, the exact use and conditioning of such elements is a disputed topic in linguistics. In this paper we address the issue of predicting the use of linking elements in Danish. Following previous research that shows how the choice of linking element might be conditioned by phonology, we frame the problem as a language modeling task: Considering the linking elements *-s*/-∅ the problem becomes predicting what is most probable to encounter next, a syllable boundary or the joining element, *s*. We show that training a language model on this task reaches an accuracy of 94 %, and in the case of an unsupervised model, the accuracy reaches 80 %.

1 Introduction

In Danish, Norwegian and Swedish, as well as in other Germanic languages, a common way of forming new words is by compounding. Here, novel words can be formed by combining already known words with an addition of a linking element between the components. Within linguistic research this linking element is somewhat of a puzzle: First of all, several languages within the Germanic family seem to share similar linking elements (Fuhrhop and Kürschner, 2014). The origin of these elements are however disputed (Nübling and Szczepaniak, 2013). Even if we assume a common origin, the use and distribution of single elements have changed among daughter languages, and we often find contradicting examples e.g., when comparing Ge. *Volk-s-musik* and Da. *folk-e-musik* 'folk music' (Fuhrhop and Kürschner, 2014). Secondly, even though the choice of a linking element may be clear to the individual speaker, linguists still struggle to establish rules for when the individual elements occur. In Danish, the linking element is decided from the first member of a compound. But when looking for rules that systematize what words take which element, only few guidelines are given (Hansen and Heltoft, 2011). Interestingly, recent studies on linking elements in German suggest that the choice of linking element is at least partially phonological, determined by features such as stress (Nübling and Szczepaniak, 2013).

Compounding has received attention in language technology as well, since it is the essence of one of the main challenges within this field: that language is productive. Within the area of statistical machine translation, segmentation of compounds into units is an important task, e.g., when translating compound words from German to English where compounding is not productive (Sag et al., 2002). Similarly, when translating English multiword expressions (MWE) into German, methods for synthesis or generation of compounds are called for (Stymne et al., 2013). Here the choice of a correct linking element becomes an issue. In the work by Cap and Fraser (2014) they rely on a rule-based morphological analyzer for German to generate the correct compounding form. Here they report that on a reference set of 283 correctly identified compounds 44 had an incorrect linking element. Recent work by Matthews et al. (2016) proposes a translation model from English MWE to German compounds that allows for modeling linking elements. In their work, they report a high recall score when generating novel compounds. Their error analysis show that their model has issues when choosing linking elements (e.g., when generating *Kirchen*türme instead of the correct *Kirch*türme 'church towers'), but they do not further provide any metrics on this subtask.

In this paper we wish to see how well a simple character-based language model is able to predict the usage of linking elements in Danish. More specifically we will look at the case of predicting occurrence of two elements -s and -$\varnothing$ (traditionally referred to as a *nulfuge* 'zero link') in Danish noun-noun compounds.

2 The linking element in Danish

In Danish, a compound is formed by attaching a linking element to the stem of its first member. Table 1 shows a list of the most common linking elements. The choice of linking element in a compound is determined by the first member. While most nouns only have one possible linking element, we do find alternation: First of all, a noun may have more competing elements that are used in connection with certain second members. An example is the noun *båd* 'boat' to which different elements can be attached depending on the compound (*båd-skat* 'boat tax', but *båd-e-byggeri* 'boat building'). Secondly, some nouns have alternating linking elements that can be used interchangeably as in *aluminium*(*s*)*rør* 'aluminum tube'.

Hansen & Heltoft (2011) present rules only for cases where the linking element -s is used: Usually an -s occurs when the first member is a compound, but many exceptions can be added to this rule. Moreover, words derived with the suffixes -(*n*)*ing*, -*ion*, and -*tek* always get an -s.

LE	Example	
-s	*idræt*	*idrætsdag* 'sports day'
-$\varnothing$	*ankel*	*ankelled* 'ancle joint'
-e	*mælk*	*mælkepulver* 'milk powder'
-er	*student*	*studenterhue* 'graduation hat'
-n	*rose*	*rosenbed* 'rose bed'

Table 1: Linking elements (LE) in Danish. The elements above the separator are considered productive while the elements below are only found in isolated forms (Hansen and Heltoft, 2011).

In some dialects the use of -s is preferred instead of the unmarked -$\varnothing$ in some cases. This could suggest that phonology does play a role in the choice of linking element as was also proposed by Nübling and Szczepaniak (2013) in the case of German. Following this suggestion, we focus on the two linking elements -s and -$\varnothing$ in order to explore how well the choice between these elements can be predicted using a character-based language model.

3 The task

We formulate the problem of determining the correct linking element of a noun as a language modeling task over the characters of a word: Given a word as a sequence of characters, what is the most probable element to come next, assuming that the word continues? Thus, when trying to determine whether a noun should take -s or -$\varnothing$, the problem becomes to estimate what would be most probable to observe next: an *s* or a *syllable boundary*?

The intuition behind this approach is, assuming that some underlying phonological process governs the choice of linking elements, then, learning the distribution over the sounds or characters of a word (including the two linking elements—the sound s and a syllable boundary), will help us to predict what element will occur as the linking element of the first member of a compound.

Data The dataset that we use for the task is *Retskrivningsordbogen* (RO) (Jervelund, 2012), which is the main source for official orthography in Danish with over $61,000$ entries in total. In RO, all words are marked with syllable boundaries (indicated by a $+$ in the text string) and information on what linking element(s) the word takes as a first member of a compound. From RO we extract the syllabified forms of nouns with linking element -s/-$\varnothing$ (excluding nouns with alternate linking elements), providing us with a dataset of 6,880 instances of nouns and linking elements.

4 Experiments

We introduce two models to approach the problem and two baselines from which we make our conclusions. First, we investigated whether character-based language models would be able to estimate the correct linking element of a word. To this end, we trained a language model on syllabified words together with their linking element ($\mathtt{s}$/+). Second, we approached the problem in an unsupervised manner, training a general language model on syllabified words, without providing any specific information on linking elements.

In both experiments the models are evaluated as a prediction task on how well they are able to predict the correct linking element of a word by weighing the estimated probabilities of observing an s or a syllable boundary ($+$) in the end of

Training objective	Input	Training signal	Prediction task	Correct answer
Language model, unsupervised	ˆi+dræt		$P(s) > P(+)$?	True
Language model, unsupervised	ˆan+kel		$P(s) > P(+)$?	False
Language model, supervised	ˆi+dræt$	s	$P(s) > P(+)$?	True
Language model, supervised	ˆan+kel$	+	$P(s) > P(+)$?	False

Table 2: Examples of input, training signals and prediction task for the supervised and unsupervised approaches. Adapted from (Linzen et al., 2016).

a word. In order to validate the performance of our models, we employ 5-Fold Cross-Validation on the set of -*s*/-∅ nouns from RO. For each iteration, we train a model with four folds and we divide the remaining fold equally for development and test.

4.1 Experiment 1: Supervised approach

In the first experiment, we train a character-based Recurrent Neural Network (RNN) language model on the entire set of -*s*/-∅ nouns from RO, including the linking element at the end of each instance.

We use a two-layer RNN with LSTM that receives an embedded representation of the characters with 128 dimensions, which are learned while training. Each LSTM layer has 64 dimensions and predictions are made using a softmax over the vocabulary of characters. We train the model using Stochastic Gradient Descent with cyclical learning rate (Smith, 2015) using the DyNet framework (Neubig et al., 2017).

As Table 2 shows, besides a beginning-of-word symbol (ˆ), we also add an end-of-word symbol (EOW) ($) to the input. We expect that this addition will improve the performance of the model, as it helps to supervise the training signal more clearly by restricting the distribution of s and $+$ as compounding elements to occur only after the EOW symbol. However, this approach also adds noise to the signal as the original sequence of characters is altered.

4.2 Experiment 2: Unsupervised approach

In the second experiment, we train an RNN identical to that in the first experiment, but without including any specific information on linking elements. Thus, at test time, this model has not been trained on nouns and linking elements, but would estimate the probabilities, $P(s)$ and $P(+)$, from the distribution of s and $+$ word-internally. The model is trained on the words from RO that are not included within the set of -*s*/-∅ nouns. The difference between the two models is summarized in Table 2.

4.3 Baseline

For each of the experiments we create two baselines. The first baseline common to both experiments chooses the most frequent linking element from the dataset. In the supervised approach, this means choosing the most frequent label from the training set. In the unsupervised case, this corresponds to the most frequent character (as it does not have access to labeled examples). In the second baseline we create an iterative back-off model that attempts to match the input word with already observed sequences of syllables from the training set.

For the supervised model, the reason that we create this baseline is because we know the rime of a word may be predictive for the choice of linking element. Thus, the back-off model starts by trying to retrieve the whole word in order to test if this was observed during training. If not, it will try to match iteratively shorter sequences of syllables until a matching rime is found. If a match is found the most frequent case of linking element is predicted. If no match is encountered, the model will back-off to the most frequent strategy.

The back-off model in the unsupervised case is similar. Here, the only difference is that we do not look for rimes, but all of the possible continuous subsequences of syllables. This is done in order to test how well a model performs in determining the joining element by remembering exact sequences of possible syllables word internally.

5 Results

The results from the two experiments are presented in Table 3. Starting with the results from Experiment 1 using the supervised approach, we see that the supervised LM reaches 0.94 for both accuracy and f1, which is higher than both of the baselines we provided. Looking more closely

	Set	Support	Supervised LM avg	std	Baseline I avg	std	Baseline II avg	std
accuracy	all	3440	**0.94**	0.009	0.56	0.023	0.83	0.023
f1			**0.94**	0.009	0.36	0.009	0.81	0.023
accuracy	seen	2977	**0.95**	0.009	0.62	0.019	0.93	0.013
f1			**0.94**	0.009	0.38	0.007	0.92	0.014
accuracy	unseen	463	**0.90**	0.024	0.17	0.028	0.17	0.028
f1			**0.82**	0.050	0.15	0.021	0.15	0.021
f1 (-∅)	unseen	383	**0.94**	0.014	0.00	0.000	0.00	0.000
f1 (-*s*)		80	**0.71**	0.088	0.29	0.042	0.29	0.042

	Set	Support	Unsupervised LM avg	std	Baseline I avg	std	Baseline II avg	std
accuracy	all	3440	0.80	0.011	0.44	0.023	**0.82**	0.001
f1			0.80	0.011	0.30	0.011	**0.82**	0.001
accuracy	seen	3287	0.80	0.010	0.41	0.022	**0.82**	0.001
f1			0.80	0.010	0.29	0.011	**0.82**	0.001
accuracy	unseen	153	0.87	0.061	**0.88**	0.035	**0.88**	0.035
f1			**0.66**	0.126	0.47	0.010	0.47	0.010
f1 (-∅)	unseen	135	**0.96**	0.035	0.94	0.020	0.94	0.020
f1 (-*s*)		18	**0.39**	0.219	0.00	0.000	0.00	0.000

Table 3: Results for the supervised and unsupervised approaches and their baselines.

into the results, we divide the test instances into two subsets, `seen` and `unseen` words, indicating whether words with the same rime were found during training. Considering the `seen` words, the LM only has a small gain compared to Baseline II, which was the baseline that used observed rimes to determine the linking element of a word. Contrarily, if we observe the set of `unseen` words, the gain is much higher. However, this set of words is imbalanced with respect to what linking elements are represented. This is reflected in the low accuracy score of 0.17 of both baselines, that in these cases choose the most frequent linking element observed in the training set (*s*). If we compare the f1 score for this set of words to the f1 score of the `seen`, the performance is lower. This is due to the model being worse at predicting the occurrence of -*s* in the `unseen` examples where it only reaches an f1 score of 0.71.

Turning to the results of the second experiment, Baseline II clearly outperforms Baseline I except for in the unseen cases, where the two baselines have the same strategy of choosing the most frequent of the characters in the training data. Furthermore, we observe that the unsupervised LM performs similarly to Baseline II overall. In the specific case of the unseen words, we can observe that the f1 score is moderately higher. Here the model does find a strategy of predicting a joining element (in contrast to the two baselines that always choose -∅), however, the f1 score of -*s* is still quite low. This is similar to the behavior of the supervised model on its unseen test instances. However, the individual results for -*s* in these cases are supported by relatively few instances (80 and 18 examples in the supervised and unsupervised experiments respectively) which is also reflected in high standard deviations.

6 Discussion

By providing a character-based language model with tagged data consisting of words and their joining elements, the model performs well on the test set. This is the case for words that are similar to the ones observed during training. But also, the model is able to generalize to words with previously unseen structure.

In the unsupervised approach, in which we did not provide any information on joining elements, the model still performs well. However, it does not outperform the baseline that retrieves sequences observed while training. This means that we cannot say that the representations learned by this model are more powerful than simply recalling ob-

served sequences. Nevertheless, the model is able to predict the joining element in some cases of unseen rimes.

7 Conclusion & future work

In this paper we approached the issue of predicting the linking element of Danish -*s*/-∅ compounds using a character-based language model. When using a language model trained of examples of words and linking elements, we reach an accuracy of 94 %. Using a language model that has never seen tagged examples reaches an accuracy of 80 % on the same task. These are promising results, but we need further error analysis to better understand the examples in which language modeling is struggling to identify the correct elements.

To pursue the approach of language modeling further, one future line of work would be to add more information to the training signal. As mentioned in the introduction, features such as stress may be an important factor in the phonological processes determining what linking element is chosen. Such information is not immediately apparent using the orthographic representation of a word as was used in this experiment. In this respect, it would be interesting to see how the models perform using phonetic transcriptions instead. Since such transcription is expensive, one could try to construct this level using grapheme-to-phoneme conversion software. As an alternative one could also attempt to reproduce the experiment using speech data.

In this paper we used a dictionary of words as training corpus. An alternative would be to use a collection of text in which information about word frequency would be included. This, in turn, might result in a different model that would be interesting to compare to the one presented above.

Furthermore, it would be interesting to see how well this approach is able to predict other linking elements in Danish, as well as in other languages.

Acknowledgments

The first author is supported by the project *Script and Text in Time and Space*, a core group project supported by the Velux Foundations. We are grateful to Patrizia Paggio for her support and comments regarding this paper. We would also like to acknowledge the anonymous reviewers for their suggestions and comments.

References

Fabienne Cap, Alexander Fraser, Marion Weller, and Aoife Cahill. 2014. How to produce unseen teddy bears: Improved morphological processing of compounds in SMT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 579–587, Gothenburg, Sweden. Association for Computational Linguistics.

Nanna Fuhrhop and Sebastian Kürschner. 2014. Linking elements in Germanic. In Susan Olsen Franz Rainer Peter O. Müller, Ingeborg Ohnheiser, editor, *Word formation. An international handbook of the languages of Europe*, HSK 40/1, page 568–582. de Gruyter Mouton, Berlin/New York.

Erik Hansen and Lars Heltoft. 2011. *Grammatik over det Danske Sprog*, 1 edition, volume 1-3. Syddansk Universitetsforlag.

Anita Ågerup Jervelund. 2012. *Retskrivningsordbogen*, 4. udg. edition. Alinea, København.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Austin Matthews, Eva Schlinger, Alon Lavie, and Chris Dyer. 2016. Synthesizing compound words for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1085–1094, Berlin, Germany. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Damaris Nübling and Renata Szczepaniak. 2013. Linking elements in german origin, change, functionalization. *Morphology*, 23(1):67–89.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Leslie N Smith. 2015. 'cyclical learning rates for training neural networks', cite. *arXiv preprint arxiv:1506.01186*.

Sara Stymne, Nicola Cancedda, and Lars Ahrenberg. 2013. Generation of compound words in statistical machine translation into compounding languages. *Computational Linguistics*, 39(4):1067–1108.