Computational Linguistics in the Netherlands Journal 5 (2015) 49-64 Submitted 06/2015; Published 11/2015

Expanding n-gram training data for language models based on morpho-syntactic transformations

Lyan Verwimp LYAN.VERWIMP@ESAT.KULEUVEN.BE
Joris Pelemans JORIS.PELEMANS@ESAT.KULEUVEN.BE
Hugo Van hamme HUGO.VANHAMME@ESAT.KULEUVEN.BE
Patrick Wambacq PATRICK.WAMBACQ@ESAT.KULEUVEN.BE

KU Leuven, Leuven, Belgium

Abstract

The subject of this paper is the expansion of n-gram training data with the aid of morpho-syntactic transformations, in order to create a larger amount of reliable n-grams for Dutch language models. The main aim of this technique is to alleviate a classical problem for language models: data sparsity. Moreover, since language models for automatic speech recognition are usually trained on written language resources while they are tested on spoken language, certain patterns that are typical for spontanous spoken language will be under-represented and patterns characteristic of written language will be over-represented. By adding transformed n-grams, we hope to adapt the language model such that it matches better with spoken language. We investigate whether a language model trained on the expanded data performs better than a baseline n-gram model with modified Kneser-Ney smoothing in terms of perplexity and word error rate. Several alternatives for the probability estimation of the transformed n-grams are explored, and an approach to deal with separable verbs in Dutch is also discussed.

1. Introduction

Statistical language models are used in many areas of natural language processing such as speech recognition, machine translation and information retrieval. They serve to determine the likelihood of a word sequence, which in the case of speech recognition, helps to distinguish between acoustically similar utterances. Classical language models, so-called *n-gram* models, predict the probability of a word given the $n-1$ previous words. These models have proven to be efficient and easy to apply in a wide variety of tasks, hence they are still widely used. However, n-gram models suffer from certain deficiencies, among others data sparsity and the fact that they are usually trained on written language, whereas they should model spoken language for speech recognition. The aim of this paper is two-fold: we attempt to alleviate the sparsity problem of language models and try to adapt a language model trained on written language such that it will better model patterns typical of spoken language.

We propose syntactic and morphological transformations of Dutch n-grams to create a larger amount of reliable n-grams based on the same training text. In comparison with English, Dutch has a more variable word order (although governed by syntactic rules) and a richer morphology, thus it is more likely to be subject to data sparsity. We try to overcome this variability by constructing new n-grams that are syntactic and morphological transformations of n-grams that occur in the training text. Hereby we mainly target n-grams that occur with a different relative frequency in written language and in spoken language. In that sense our approach is also an attempt to adapt the model to the domain of the application (ASR), namely spontaneous spoken language.

Recently, *recurrent neural network* based language models (*RNNLMs*) (Mikolov et al. 2010) have proven to perform better than n-gram models, but training them is computationally much more complex. Moreover, we are not aware of the use of an RNNLM in the first pass of a speech recognizer except for Hori et al. (2014), but their models are trained on only 1.1M words and have a vocabulary of 47k words, which is rather small for large vocabulary speech recognition (our language models are built with a vocabulary of 400k words). In multi-pass decoding, one uses an efficient and fast language model such as an n-gram in the first pass to reduce the search space, after which a more complicated model such as an RNNLM can be used in the second pass to rescore n-best lists (Mikolov et al. 2010) or word lattices (Liu et al. 2014). Multi-pass decoding usually improves the recognition performance, but also slows down the decoding process compared to one-pass decoding – which corresponds to only performing the first pass of the multi-pass process and choosing the single best hypothesis rather than a set of best hypotheses. Since n-gram models are more efficient to train and easier to use than RNNLMs, and since Mikolov et al. (2010) show that interpolating an RNNLM with an n-gram model works even better than only using the RNNLM, we believe that work on n-gram models is still useful.

The remainder of this paper is organized as follows: section 2 explains the idea of morpho-syntactic transformations; section 3 compares this idea to related work; section 4 tests the validity of our work experimentally; and finally, we end with a conclusion in section 5.

2. Morpho-syntactic transformations

In this section, we first clarify the concept of syntactic and morphological transformations (sections 2.1 and 2.2) and the way in which we apply them to the original n-grams (section 2.3). Next, we explain how we estimate the probability of the new transformed n-grams (section 2.4). Finally, some issues that our approach encounters are discussed (section 2.5).

2.1 Syntactic transformations

In many languages, the word order in certain syntactic patterns can be reversed. We explore three types of these syntactic transformations. Firstly, the default order of *subject - verb (SV)* in Dutch (1) can be reversed (*VS*) in a question (2) or if another word occupies the first position in the sentence (3):

(1) ***Hij eet*** *een boterham.* “He eats a sandwich.”

(2) ***Eet hij*** *een boterham?* “Does he eat a sandwich?”

(3) *Daarom* ***eet hij*** *een boterham.* “That is why he eats a sandwich.”

We expect that questions such as (2) occur more often in spoken language than in written language since the interaction between people is more prominent in the former. If we compare the statistics for the *SV* and *VS* patterns in corpora of written and spoken language, we see that this is indeed the case. In table 1, one can see the number of sentences that contain a certain pattern in a parsed corpus of written language (*LASSY – Large Scale Syntactic Annotation of written Dutch* (Van Noord 2006, Van Noord 2008)) versus a parsed corpus of spoken language (*CGN – Corpus of Spoken Dutch* (Oostdijk 2000)), obtained with the help of *GrETEL (Greedy Extraction of Trees for Empirical Linguistics)*, a tool with which one can query treebanks based on a natural language example (Augustinus et al. 2012). As the first two rows show, the percentage of *VS* is higher in spoken language than in written language, and the opposite is true for *SV*.

Secondly, the order of *verb - direct object (VO)* in a head clause (4) is switched to (*OV*) in a subordinate clause (5):

(4) *Hij* ***schrijft een boek.*** “He writes a book.”

(5) *Het klopt dat hij* ***een boek schrijft.*** “It is correct that he writes a book.”

		written language		spoken language	
pattern		# of sentences	percentage	# of sentences	percentage
syntactic	*SV*	31,394	**48.15**	42,081	32.39
	VS	1,868	2.87	10886	**8.38**
	VO	10,789	**16.55**	15,550	11.97
	OV	5,073	**7.78**	5,016	3.86
morph	*1sg*	495	0.76	9,194	**7.08**
	1pl	427	0.65	2,037	**1.57**
	2pl	4	0.01	83	**0.06**

Table 1: Frequency of several syntactic and morphological patterns (the number of sentences in which 1 or multiple matches are found) in written language versus in spoken language. The corpus of written language consists of 65,200 sentences and the corpus of spoken language consists of 129,923 sentences.

Our training text consists of newspaper material, which means that it will contain many more subordinate clauses than conversational language does (see among others O'Donnell (1974) and Akinnaso (1982)). The statistics in table 1 confirm this: although in general, people use more sentences with a verb and a direct object in written language, the proportion of *VO* patterns with respect to *OV* patterns is 2.1 for written language and 3.1 for spoken language, meaning that people use relatively more *VO* than *OV* in spoken language compared to written language. By transforming the word order of n-grams that only occur in subordinate clauses to the word order of a head clause, we hope to cover more possible word sequences in *VO* order. Although we expect the largest improvement for $SV \rightarrow VS$ and $OV \rightarrow VO$ transformations, the transformations are applied in both directions.

A last example of a syntactic pattern that can be reversed is a conjunction, in which the different components can usually be exchanged:

(6) ***De jongen en het meisje*** *gingen wandelen.* "The boy and the girl went for a walk."

(7) ***Het meisje en de jongen*** *gingen wandelen.* "The girl and the boy went for a walk."

Note that this is not the case when the conjunction is a fixed phrase or idiom, e.g. *zonder slag of stoot* "without a shrug", literally "without a hit or a push". Since most of the collected conjunctions were longer than three words (which is the maximum n-gram order that we will use in this paper), the reversed conjunctions will not be added to our final language model.

2.2 Morphological transformations

The morphological transformations that are added to the language model are also motivated by the need to bridge the gap between written and spoken language: in conversational language, the first person and second person are used much more often, whereas newspaper articles usually talk about other people and thus use more third persons. The last three rows in table 1 confirm this intuition. Hence, if we can transform third person n-grams into first and second person n-grams, we are potentially improving the model to cover spoken language.

In Dutch, the plural forms of a verb are the same for first person (8), second person (9), third person (10) and infinitive (11):

(8) *Wij* ***luisteren****.* "We listen."

(9) *Jullie* ***luisteren****.* "You listen."

(10) *Zij* ***luisteren****.* "They listen."

(11) *Zij houden van* ***luisteren****.* "They like to listen."

The first and second person of the plural can thus easily be constructed by combining *wij* "we" and *jullie* "you" with a present plural or an infinitive. For the first person singular (in the present tense) however (12), the verb form is different and only corresponds to the (infrequent) imperative (15):

(12) *Ik* ***luister****.* "I listen."

(13) *Jij* ***luistert****.* "You listen."

(14) *Hij* ***luistert****.* "He listens."

(15) ***Luister!*** "Listen!"

As the examples above show, the conjugation of the first singular can be constructed by removing the *-t* suffix from the second (13) or third singular (14) of the present. There are some exceptions to this rule, but these concern very frequent irregular words or can be captured with an additional rule. Therefore the first person singular can also be added with the help of a rule-based transformation.

In this paper, we will limit ourselves to *SV* - *VS*, *VO* - *OV*, first singular (*1sg*), first plural (*1pl*) and second plural (*2pl*) present tense transformations. Although other transformations are possible (e.g. past tense), analysis of the speech recognition results of the baseline language model on our development set shows that these transformations already have the potential to improve recognition accuracy: 15% of the words that are recognized incorrectly are verbs, of which 14% are first person singulars of the present and 14% are plural forms of the present.

2.3 Transforming *n*-grams

The *n*-grams in the training data are transformed as follows: for the syntactic transformations, the data is parsed with a dependency parser (see section 4.1). Each node in the parse tree contains a lot of information such as the part-of-speech (POS), the dependency relation, the lemma, the begin and end position of the constituent, etc. We extract only the words themselves, their POS tags and their dependency relations. If a phrase consists of more than one word, it is the parent node that carries the dependency relation of the whole phrase e.g. in diagram 1 the parent node of *een boek* has the relation *direct object*, but the nodes of *een* and *boek* do not. This means that in principle we have to recursively search for all children, grandchildren, great-grandchildren and so on that belong to the phrase carrying a particular dependency relation. For example, if the sentence would be "hij schrijft een boek dat de wereld zal verbazen" (he writes a book that will astonish the world), *een boek dat de wereld zal verbazen* is the direct object but the words of *dat de wereld zal verbazen* belong to grandchildren, great-grandchildren etc. of the *direct object* node because *dat de wereld zal verbazen* is a relative clause modifying *een boek*. However, since going deeper than the children usually returns word sequences that are longer than the maximum *n*-gram order we use, we only include the children. We use the POS and the dependency relation to search for the correct patterns: a word or constituent with the relation *subject* followed or preceded by a word with the relation *head* that has a verbal POS together constitute an *SV* or a *VS* pattern, and a word or constituent with the relation *direct-object* followed or preceded by a *head* with a verbal POS form an *OV* or a *VO* pattern.

The morphological transformations are achieved by POS tagging and lemmatizing the training corpus (see section 4.1 for information about the POS tagger). We extract the verb conjugations that are tagged as second/third person singular of the present tense or as infinitive. The first person singular forms are then constructed – if not yet present in the corpus – based on some simple rules:

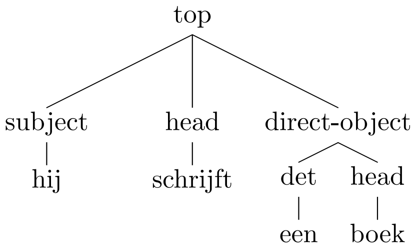

Diagram 1: Dependency parse tree for the sentence "hij schrijft een boek"

		syntactic		morphological			
n-gram order	number of n-grams in baseline model	SV	OV	1sg	1pl	2pl	total added
1	400,000	-	-	1,960	-	-	1,960 (0.49%)
2	11,743,568	516,293	135,751	2,798	6,798	7,571	669,211 (5.70%)
3	41,593,463	876,163	301,085	-	-	-	1,177,248 (2.83%)
total	194,097,138	2,223,576	691,929	2,798	6,798	7,571	2,936,189 (1.51%)

Table 2: Number of new n-grams per n-gram order.

normally, the *-t* suffix is removed from the second/third person singular. For certain exceptions, the lemma is needed: if the lemma ends on *-ten*, the *t* is part of the stem and is not removed (e.g. *moeten* → *moet*). If the lemma ends on *-aan*, *-at* is removed (e.g. *staan*, *staat* → *sta*). The first and second person plural bigrams are constructed by combining *wij* and *jullie* with the infinitive. We chose to use only the infinitives because these are much more frequent than the present plurals, and because the employed POS tagger seems to make more mistakes for the present plurals than for the infinitives.

All patterns corresponding to the discussed transformations are collected. Table 2 gives an overview of the amount of new n-grams that are created by applying each of these transformations to our training data. Only for the first person singular, new unigrams are added because this is the only transformation for which new word forms are created. The morphological transformations only consist of a pronoun and a verb, thus no new trigrams are added for this type of transformation. It can be seen that the number of syntactic transformations is much higher than the number of morphological transformations.

2.4 Probability estimation

There are two main possibilities to add the transformed n-grams to the baseline language model: we can either perform a count-based probability estimation or we can directly integrate the new n-grams in the language model. The count or probability that is assigned can also be determined in multiple ways. The transformed n-grams are only added if they do not occur in the corpus.

If we use a count-based estimation of the probabilities, the simplest thing to do would be to assign a count of 1 to all new n-grams (this type of model will be referred to as *one* in the remainder of the paper):

$$C_{new} \leftarrow 1 \tag{1}$$

If we base the count of the new n-gram on the count of the original n-gram, there are two possibilities: in the first approach, the count C_{orig} of the original n-gram is not changed and the count C_{new} of the new n-gram is assigned a fraction (indicated by α) of it:

$$C_{new} \leftarrow \alpha C_{orig} \tag{2}$$

A special case in this scenario occurs when $\alpha = 1$ and thus the original and new count are the same (henceforth called *same*). The models in which this $\alpha = 0.33$ will be referred to as *third*.

Alternatively, the count of the original n-gram is re-distributed over the original and the new n-gram:

$$\begin{aligned} C_{new} &\leftarrow \alpha \\ C_{orig} &\leftarrow C_{orig} - \alpha \end{aligned} \tag{3}$$

If $\alpha = 1$ in the above equation, the extended language model will be called *distr-1*. In order to separate the effect of discounting the original n-grams and adding the new n-grams, this model will be compared to one in which only the original n-grams are discounted, so the new n-grams are *not* added (*disc-1*) (this corresponds to equation (3) in which $\alpha = 0$).

Using the new n-grams and the adapted counts, we then train a new language model for each n-gram order and linearly interpolate them. This is necessary because we do not have higher order $(n+1)$-grams corresponding to the newly added lower order n-grams e.g. adding the new bigram *ik luister* from (12) to a 3-gram model would be invalid without a corresponding 3-gram (e.g. *ik luister naar*), which is difficult to create because there are in principle infinitely many 3-gram extensions for (12), the majority of which is ungrammatical. Since the baseline 4- and 5-gram models did not show any significant improvements in terms of perplexity or word error rate with respect to the 3-gram model, we will only discuss 3-gram language models.

Both in *one*, *same* and *third* on the one hand and *distr-1* on the other hand, the final probability of the original n-grams decreases: in the latter case this is quite obvious and in the former case we increase the total amount of n-grams such that the probability of the original n-grams – of which the count is not changed – decreases. Since it is not clear which of the two strategies is better, we will investigate both.

With respect to adding the transformed n-grams directly to the language model, we will not further investigate this possibility since the probabilities would have to be re-normalized in order to test the perplexity on a test set, and since a change in the counts naturally leads to a change in the probabilities.

2.5 Issues

Our approach faces several issues which might influence the performance of the adapted language models. A first issue is specific to Dutch (and a few other languages such as German): separable verbs can complicate the process of generating syntactic transformations. Separable verbs are verbs that consist of two parts that are split in a head clause (16), but merged in subordinate clauses (17):

(16) *Zij* ***lossen*** *dat* ***op***. "They fix it."

(17) *Het klopt dat zij dat* ***oplossen***. "It is correct that they fix it."

Reversing the word order of an *OV* or *VO* sequence containing a separable verb would thus lead to ungrammatical n-grams e.g. *dat oplossen* would become **oplossen dat* and *lossen dat op* would become **dat lossen op*. Ideally, the separable verbs are detected and either split ($OV \rightarrow$ *VO*) or merged (*VO* $\rightarrow OV$). Unfortunately, the employed parser only detects a separable verb if it is already separated. The *VO* n-grams that are found can thus be transformed and corrected by merging the particle with the finite verb. For the OV patterns however, we need another strategy to detect whether the verb is separable or not. One possible way to detect separable verbs is to use a morphological analyzer. However, there also exist verbs in Dutch that consist of multiple morphemes but that are not separable (e.g. *betalen*: *be-talen* 'to pay'), so this strategy would wrongly split non-separable verbs. Another possibility is using a list of separable verbs and their conjugations: from *Wiktionary* (2015) we extracted a list of separable verbs in Dutch. Unfortunately, the *Wiktionary* entries of the verbs do not always provide their conjugations, or they sometimes provide conjugations that are obsolete and/or rarely used. That is why we also make use of a list of lemmas and their inflected forms, extracted from the *SoNaR* corpus (Oostdijk et al. 2013). If a verb in an *object - verb* sequence is separable, it is split while the constituent order is reversed (to *verb - object*). The vocabulary is not changed, so it is possible that some of the newly split verbs are out-of-vocabulary. We discuss the results of the splitting of separable verbs in section 4.2.

A second issue is the fact that the dependency parser and POS tagger sometimes introduce incorrect parses/tags, which can lead to incorrect new *n*-grams. Since our training data consists of newspaper articles, which typically contain complicated and long sentences, the probability of parsing errors is considerable. The inspection of random parses that were produced confirms this:

(18) *Over de vluchtroute is voorlopig niets bekend.* "For now, nothing is known about the escape route."

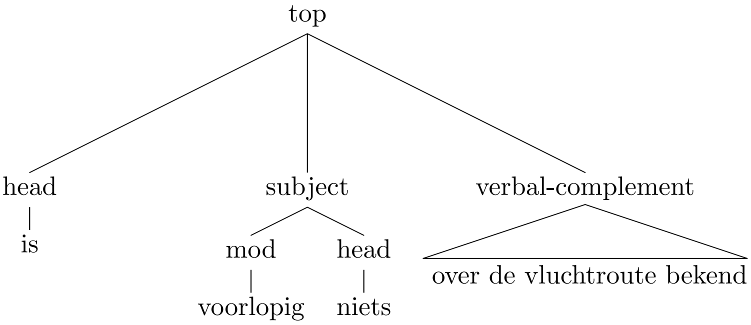

Diagram 2: Dependency parse tree for the sentence "Over de vluchtroute is voorlopig niets bekend."

In (18), the parser incorrectly classifies *voorlopig niets* as the subject of the sentence: *voorlopig* is interpreted as the modifying adjective of *niets* (see the parse tree in diagram 2), whereas it is an adverb modifying the verb phrase. Moreover, since dependency parsing puts the head of a phrase in front and since the head of a sentence is the finite verb, the verb is moved to the front of the parse, resulting in (19).

(19) *is voorlopig niets over de vluchtroute bekend*

These word order changes occur quite frequently because in Dutch, the finite verb of a subordinate clause occurs at the end rather than at the beginning of the clause. Therefore we only add transformed *n*-grams to the model if the word order in the corresponding original *n*-gram is not changed by the parser, because reversing the word order in an *n*-gram in which the word order is already changed will most likely lead to ungrammatical *n*-grams.

As we already briefly mentioned in section 2.3, the POS tagger also makes mistakes. Words ending with suffixes that could be verbal suffixes, such as *clauwaert*, *vandeneynde* and *paars-witte* are often tagged as verbs. We use several heuristics in order to reduce the number of errors, such as ignoring words with archaic orthography (which is exclusively used in proper nouns such as *ae* in *clauwaert* and *ey* in *vandeneynde*), ignoring words starting with a capital and only including the verbs that are tagged with high confidence.

3. Related work

The integration of syntactic information in language models has been done before by conditioning a word not on the previous $n-1$ words, but on its neighbors in a dependency parse (Pauls and Klein 2012, Gubbins and Vlachos 2013, Sidorov et al. 2014, Zhang and Wang 2014) or jointly training on word sequences and dependency parses (Chelba et al. 1997, Chelba 1997). However, for these approaches the test set has to be parsed too, which is extremely time-consuming so not suitable for real-time speech recognition. Our research on the other hand, focuses on using the syntactic information but encoding it in a normal n-gram model, which is fast and efficient. Morphological information has also been used before, in for example *factored language models* (Bilmes and Kirchhoff 2003, Vergyri et al. 2004, Alumäe 2006), but these are more complex than n-gram models and cannot – as far as we know – be used in one-pass decoding for speech recognition. We try to use syntactic and morphological knowledge to create n-grams in such a way that our test set does not have to be parsed and we can easily use our language model in the first pass of speech recognition.

Since we increase the amount of n-grams for the same training text, our approach can also be described as data expansion, which has already proven to be useful for statistical machine translation (Tyers 2009, Sánchez-Cartagena et al. 2011).

Finally, Schukat-Talamazzini et al. (1995) and Vandeghinste (2009) have shown that language models in which the word order in n-grams can be changed – which is what we propose to do in a syntactically motivated way – work better than classical n-gram models.

4. Experiments

4.1 Data sets and tools

Our baseline language model is trained on normalized extracts from the Flemish newspaper *De Standaard*, which contain approximately 128M word tokens. This is a middle-sized corpus for language modeling (sometimes billions of words are used), but that is no issue since the purpose of our research is to cover more grammatical sequences with the same amount of data such that eventually, we need only a middle-sized corpus in order to train a proper language model. We use two test sets for perplexity evaluations: a first corpus containing only inversion questions and a second corpus consisting of transcriptions of spontaneous speech.

The first test set contains 248 sentences (3165 word tokens) constructed with the aid of *GrETEL* (Augustinus et al. 2012). Since our training set consists of newspaper material which typically does not contain a lot of questions, we expect that more *VS* (in comparison with *SV*) patterns will be added to the new model, and consequently the extended model should work better on a test set consisting of inversion questions.

The second test set consists of 20 transcriptions of fragments of component *g* of *CGN*) (Oostdijk 2000), containing the transcripts of discussions, debates and meetings (*CGN-g-test*: ca. 1h 52min or 2511 word tokens). As a validation set to test all the set-ups and determine the interpolation weights for the models of different n-gram order, we use another 20 fragments of component *g* of *CGN* (*CGN-g-dev*: ca. 2h 53min or 4381 word tokens). For speech recognition experiments, we use the same validation and test sets of *CGN* that are used for perplexity evaluations. We first test all set-ups on the validation set (section 4.2 and section 4.3) and then on the test set (section 4.4). Finally, in section 4.5 we will look more closely at our extended language models and the results of the evaluation.

The training corpus was parsed with *Alpino*, a dependency parser for Dutch (Van Noord 2006), and POS tagged and lemmatized with *Frog* (Van den Bosch et al. 2007). The *SRILM* toolkit (Stolcke 2002) was used to train language models and compute the reported perplexities. The speech recognition experiments were done using the *SPRAAK* toolkit (Demuynck et al. 2008), configured according to Demuynck et al. (2009). All the language models were trained using a vocabulary consisting of the 400k most frequent words in the training set (the full vocabulary contains 1.1M

test set	pattern	baseline	one	same	third	distr-1	disc-1
inversion questions	both	276.9	277.3	277.4	277.3	**276.6**	**276.6**
	SV only	276.9	277.1	277.0	277.1	**276.6**	**276.6**
	OV only	276.9	277.0	277.1	277.1	277.1	**276.6**
CGN-g-dev	both	**217.6**	218.5	218.4	218.6	**217.6**	**217.6**
	SV only	**217.6**	218.3	218.6	218.3	**217.6**	**217.6**
	OV only	**217.6**	217.8	218.0	217.8	**217.6**	**217.6**

Table 3: Perplexity results for the baseline 3-gram model versus the adapted language models with different counts for the syntactically transformed n-grams, evaluated on the corpus of inversion questions and *CGN-g-dev*. The two types of syntactic transformations, *SV - VS* and *OV - VO*, are tested separately and jointly.

	same	third	distr-1
no separation	214.43	214.55	213.64
separation	**214.40**	**214.54**	**213.63**

Table 4: Perplexity results for adapted language models with non-separated versus separated verbs, tested on *CGN-g-dev*.

word types) with interpolated modified Kneser-Ney smoothing (Chen and Goodman 1999). No count cut-offs were applied.

4.2 Perplexity experiments

In this section, we report the perplexity of the extended language models on the inversion test set and on *CGN-g-dev*. Different ways of assigning a count or probability to the transformed n-grams are explored. Given that the transformed n-grams did not occur in our training text, the count assigned to them will be either equal to or smaller than the count of their corresponding original n-gram (which is the non-reversed n-gram for the syntactic transformations, the unigram of the third person for the first person singular bigrams and the unigram of the infinitive for the first and second person plural bigrams). If the counts are fractional (in the case of *third*, where $C_{new} \leftarrow 0.33 * C_{orig}$), they are rounded off, because Kneser-Ney smoothing cannot deal with fractional counts. The results of the alternative probability estimations for the syntactic transformations are given in table 3. The extended models are all created by interpolating models of different order with optimal interpolation weights, as determined on the *CGN-g-dev* set.

On the test set of inversion questions, the extended language model in which the transformed n-grams have a frequency of 1 and the original n-grams are discounted performs slightly better than the baseline model (table 3). However, this performance may very well be due to the discounting of the original n-grams, since *disc-1* has a similar performance. There is no clear difference in models to which only one (SV or OV) of the syntactic transformations is added. For *CGN-g-dev*, neither of the language models extended with syntactic transformations improve with respect to the baseline. The best performing models are again *distr-1* and *disc-1*.

In section 2.5 we explained that separable verbs in Dutch remain one word in subordinate clauses while they are split in head clauses, and we explained the way in which we split them. The reversed n-grams with separated verbs are now tested within adapted language models with the same count, a third of the count and the re-distribution of the count (1 to the new n-gram, subtract 1 from the original n-gram). The results indicate that the splitting of the separable verbs barely influences the perplexity (see table 4).

		1st singular			1st + 2nd plural	all
syntactic	no morph	one	same	third	one	one
-	**217.6**	217.7	217.9	217.8	**217.6**	217.7
distr-1	**217.7**	**217.7**	218.0	217.8	**217.7**	217.8
third	218.6	**218.2**	218.5	218.3	**218.2**	218.3

Table 5: Perplexity results for 3-gram models with different combinations of syntactic and morphological transformations, tested on *CGN-g-dev*.

Let us now discuss the results for adding the morphological transformations to the language model (table 5). We observe a similar phenomenon as for the syntactic transformations: if only the morphological transformations are added (first row of the table), no improvement with respect to the baseline language model is obtained. If the morphological and syntactic transformations are combined, adding the morphological transformations does not improve if the syntactic transformations are added as in *distr-1*. If the syntactic transformations get a third of the frequency of the original n-grams, we see extremely small – negligible – improvements if the first person singular bigrams or the first and second person plural bigrams are added with a count of one, but this effect disappears if the model is augmented with all morphologically transformed n-grams.

4.3 Speech recognition experiments

In this section, we discuss the results of the speech recognition experiments on our development set. Table 6 shows the results for the language model extended with only syntactically transformed n-grams. For all models discussed, the differences in word error rate with respect to the baseline model are very small. There is no clear difference in performance between the two types of transformations that are added: neither SV nor OV achieves a substantial improvement.

pattern	baseline	one	same	third	distr-1	disc-1
both	30.43	30.46	30.46	30.46	30.44	**30.40**
SV only	30.43	30.42	**30.39**	30.46	30.44	30.40
OV only	30.43	30.42	30.41	30.45	30.46	**30.40**

Table 6: Word error rates for the baseline language model versus the language models extended with syntactic transformations, for different count-based estimations, tested on *CGN-g-dev*. The two types of syntactic transformations, SV - *VS* and OV - *VO*, are tested jointly and separately.

The word error rates for the morphologically augmented language models also confirm the perplexity results (table 7). If only morphological transformations are added (first row of the table), the best results are obtained for the addition of *1pl* and *2pl* bigrams. For the combination of syntactic and morphological transformations (second and third row), the models extended with *1sg* perform slightly better, but the difference is still negligible.

4.4 Test set evaluation

If we test our extended language models on a different part of component *g* of *CGN* (one which was not used to determine the interpolation weights), we see similar results for both perplexity and word error rate.

The perplexity of the extended language models, regardless of the count that is given to the new n-grams, is not better than the perplexity of the baseline model. In table 8, the results for the

		1st singular			1st + 2nd plural	all
syntactic	no morph	one	same	third	one	one + one
-	30.43	30.44	30.46	30.46	**30.42**	**30.42**
distr-1	30.44	**30.40**	30.41	**30.40**	30.43	30.44
third	30.46	30.42	**30.40**	30.43	30.42	30.44

Table 7: Word error rates for the baseline language model versus the language models extended with morphological transformations only or both morphological and syntactic transformations, for different count-based estimations, tested on *CGN-g-dev*.

models extended with syntactic transformations only are given. Again, we see that the perplexities are all very close to each other.

test set	baseline	one	same	third	distr-1
both	**213.6**	214.5	214.4	214.6	**213.6**
SV only	**213.6**	214.3	214.6	214.3	**213.6**
OV only	**213.6**	213.8	213.9	213.8	**213.6**

Table 8: Perplexity results for the baseline 3-gram model versus the adapted language models with different count-based probability estimations for the syntactically transformed n-grams, evaluated on *CGN-g-test*.

With respect to the morphological transformations (table 9), we can draw a similar conclusion: the models augmented with morphological transformations, with or without the combination with syntactic transformations, show no improvement compared to the baseline language model.

		1st singular			1st + 2nd plural	all
syntactic	no morph	one	same	third	one	one + one
-	**213.6**	213.7	213.9	213.7	**213.6**	**213.6**
distr-1	**213.6**	**213.6**	213.8	213.7	213.7	213.7
third	214.6	214.2	214.4	213.7	**213.8**	**213.8**

Table 9: Perplexity results for 3-gram models with different combinations of syntactic and morphological transformations added, tested on *CGN-g-test*.

For speech recognition, the results obtained on our development set are confirmed as well. In table 10, one can see the word error rates for the language models extended with syntactic transformations. The largest improvement in word error rate – 0.07 – is obtained with the *disc-1* model, to which no new n-grams are added. This is probably due to the fact that the discounting method applied to this model prunes infrequent n-grams (we mentioned in section 4.1 that the count cut-off is set to 1, so the model contains rare n-grams).

Finally, for models extended with morphological transformations, a similar phenomenon can be observed: the adapted language models do not perform significantly better than the baseline language model. The biggest difference with the baseline language model (0.05) is obtained by a model to which all transformations are added: the syntactic ones according to *distr-1* and the morphological ones with a count of one.

pattern	baseline	one	same	third	distr-1	disc-1
both	30.39	30.43	30.49	30.42	30.46	**30.32**
SV only	30.39	30.47	30.42	30.44	30.34	**30.32**
OV only	30.39	30.42	30.51	30.47	30.34	**30.32**

Table 10: Word error rates for the baseline language model versus the language models extended with syntactic transformations, for different count-based estimations, tested on *CGN-g-test*. The two types of syntactic transformations, *SV* - *VS* and *OV* - *VO*, are tested jointly and separately.

		1st singular			1st + 2nd plural	all
syntactic	no morph	one	same	third	one	one + one
-	30.39	30.48	30.47	30.46	**30.35**	30.39
distr-1	30.46	30.40	30.45	30.44	30.35	**30.34**
third	30.42	**30.38**	30.52	30.46	30.39	30.41

Table 11: Word error rates for the baseline language model versus the language models extended with morphological transformations only or both morphological and syntactic transformations, for different count-based estimations, evaluated on *CGN-g-test*.

4.5 Error analysis

There are several possible reasons why our approach does not give the improvements we expected: 1) the assumption that the transformed patterns occur more in spoken language than in written language is wrong; 2) the way in which the new n-grams are added is sub-optimal: the probability that we assign to them is too high or too low; 3) the added n-grams are not numerous or relevant enough to make a substantial contribution to the model and 4) the performance deteriorates for the n-grams that were in the baseline model because their probabilities decrease. We already showed in section 2 and table 1 that the first possibility is not true. As regards the second option: we tested several probability estimations in the previous sections but the probability assigned to the new n-grams did not have a great impact on the performance of the language model. In this section, we will show that the third possibility – the added n-grams are not numerous or relevant enough – is probably (one of) the reason(s) why our language models extended with transformed n-grams do not perform better than the baseline language model.

We tested how many of the n-grams that were added to the language model are actually useful during the evaluation of a test set. The results are given in table 12: for each transformation, we check how many of the bi- and trigrams in *CGN-g-dev* and *CGN-g-test* match with the transformed n-grams that are added. It appears that the amount of bi- and trigrams on which we could make a difference is extremely small. If the size of the training set decreases (see third column in table 12), the probability that a transformed n-gram is new increases. Consequently, if the training set becomes smaller, usually more relevant n-grams are added to the extended language models. However, even if the training set is 1% of the original size (1.4M words, which is quite small for language modeling), the number of relevant n-grams that can be added is negligible compared to the total number of n-grams present in the test sets. This is probably explained by the fact that for a smaller training corpus, also a smaller amount of transformed n-grams can be created.

Why is it that so few of the transformed n-grams are actually used for the evaluation on our test sets, while the number of n-grams that we add is considerable (see table 2)? We believe that our training set already contains enough relevant n-grams, and that most of the transformed n-grams that can be added will rarely be used. In figure 1, we plot the percentage of *VS* (the largest group of transformations) n-grams that are already present in the training set (y-axis on the left), relative to

test set	n-gram order	training set	relevant syntactic transformations	relevant morphological transformations	total # of n-grams in test set
CGN-g-dev	2	full	6	0	192,950
		5%	31	3	
		1%	57	12	
	3	full	7	-	154,144
		5%	18	-	
		1%	13	-	
CGN-g-test	2	full	3	0	132,947
		5%	15	1	
		1%	33	5	
	3	full	9	-	109,418
		5%	12	-	
		1%	12	-	

Table 12: Number of bi- and trigrams that were added to the extended model and that were actually present in *CGN-g-dev* and *CGN-g-test*, for different sizes of training sets (full: 128M words, 5%: 5.4M words, 1%: 1.4M words).

the frequency of the SV n-grams in our training set. On the y-axis on the right, the percentage of SV patterns that are covered is plotted: it can be seen that the SV n-grams with frequency 1 already cover 69% of the total number of SV patterns. For only 8% of those SV patterns with frequency 1, the corresponding *VS* n-gram is already present, so the majority of the *VS* n-grams that can be added (665,100, to be exact) are transformations of very rare SV n-grams and are subsequently unlikely to occur in a test set. Ideally, there are quite some SV occurrences in our training set that have a middle-high frequency and for which there is no corresponding *VS* n-gram present. However, once the frequency of the SV n-grams is 10 or higher, more than 70-80% of the corresponding *VS* n-grams are already present in the test set. Moreover, the total number of n-grams with a middle-high frequency is much smaller (the coverage graph is quite flat for frequencies 10 until 30). Thus, the 20-30% *VS* patterns that are not yet present in the training set, is relative to a much smaller total. There are for example only 244 SV n-grams with a frequency of 20 for which no *VS* transformation is present in the training set (compare this to the 665,100 for the SV n-grams of frequency 1!).

This is strong evidence that hypothesis 3 – the added n-grams are not numerous or relevant enough – is true. Moreover, since so many irrelevant n-grams are added (of which the original n-gram was very rare), it is possible that hypothesis 4 is valid as well: the irrelevant n-grams are mainly noise that possibly deteriorates the modeling of the n-grams that were already present in the training set.

5. Conclusion

We presented a method to expand n-gram training data with the aid of morpho-syntactic transformations which targets both data sparsity and domain adaptation from written to spoken language. Experimental validation shows that language models augmented with syntactic transformations based on *subject - verb* and *verb - object* order, and morphological transformations for 1st person singular and 1st and 2nd person plural of the present tense, do not work better than a baseline n-gram model with interpolated modified Kneser-Ney smoothing. The amount of n-grams that is added is too small to make a difference: only a very small percentage of the test sets match with the n-grams that we added, because the training set already contains enough relevant n-grams.

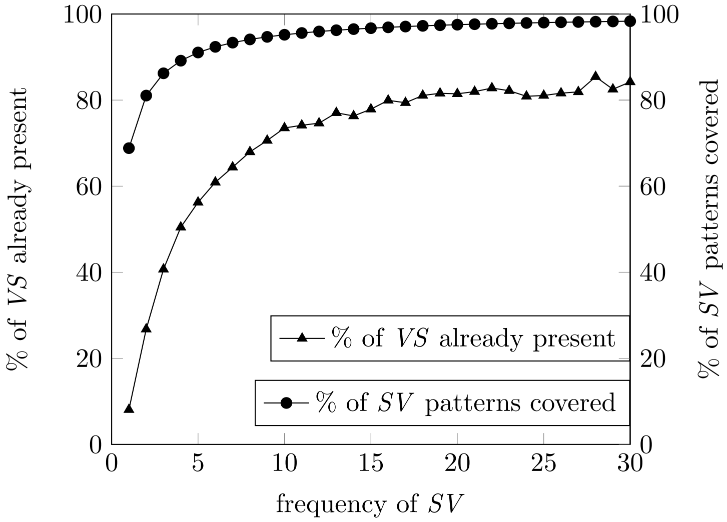

Figure 1: Percentage of transformed (SV) patterns already present in the training corpus (left) and of SV pattern coverage (right), relative to the frequency of the original SV pattern.

Although the experiments show that our approach does not work better than classical n-gram language models, it is useful to know that adding morpho-syntactic transformations to a baseline model does not work, at least not in the way that we did this. To increase the coverage, it would be possible to augment the model with even more transformations, such as the 2nd person singular of the present tense, other verb conjugations, singular and plural nouns, and so on. Nevertheless, the techniques to do this rely on the performance of other tools such as syntactic parsers and POS taggers, which are prone to errors. Moreover, adding extra transformations will probably not improve the results since it is likely that other kinds of transformations will be just as infrequent as the ones we tested.

References

Akinnaso, F.N. (1982), On the differences between spoken and written language, *Language and Speech* **25** (2), pp. 97–125.

Alumäe, Tanel (2006), Sentence-adapted factored language model for transcribing Estonian speech, *International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, pp. 429–432.

Augustinus, L., V. Vandeghinste, and F. Van Eynde (2012), Example-based treebank querying, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, pp. 3161–3167.

Bilmes, J.A. and K. Kirchhoff (2003), Factored language models and generalized parallel backoff, *Proceedings of HLT/NAACL*, pp. 4–6.

Chelba, C. (1997), A structured language model, *Proceedings of Association of Computational Linguistics - European Association of Computational Linguistics (ACL - EACL)*, Madrid, Spain, pp. 498–500.

Chelba, C., D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, and D. Wu (1997), Structure and performance of a dependency language model, *Proceedings of Eurospeech*, Rhodes, Greece, pp. 2775–2778.

Chen, S.F. and J. Goodman (1999), An empirical study of smoothing techniques for language modeling, *Computer Speech and Language* **13**, pp. 359–394.

Demuynck, K., A. Puurula, D. Van Compernolle, and P. Wambacq (2009), The ESAT 2008 system for N-Best Dutch speech recognition benchmark, *Automatic Speech Recognition and Understanding Workshop (ASRU)*, Merano, Italy, pp. 339–343.

Demuynck, K., J. Roelens, D. Van Compernolle, and P. Wambacq (2008), SPRAAK: An Open Source SPeech Recognition and Automatic Annotation Kit, *Proceedings International Conference on Spoken Language Processing*, Brisbane, Australia, p. 495.

Gubbins, J. and A. Vlachos (2013), Dependency language models for sentence completion, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1405–1410.

Hori, T., Y. Kubo, and A. Nakamura (2014), Real-time one-pass decoding with recurrent neural network language model for speech recognition, *International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pp. 6364–6368.

Liu, X., Wang. Y., X. Chen, M.J.F. Gales, and P.C. Woodland (2014), Efficient lattice rescoring using recurrent neural network language models, *International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pp. 4908–4912.

Mikolov, T., Karafiát M., L. Burget, J. Černocký, and S. Khudanpur (2010), Recurrent neural network based language model, *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pp. 1045–1048.

O'Donnell, R.C. (1974), Syntactic differences between speech and writing, *American Speech* **49** (1/2), pp. 102–110.

Oostdijk, N. (2000), The spoken dutch corpus. overview and first evaluation, *Language Resources and Evaluation Conference (LREC)*.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste, and Ineke Schuurman (2013), The construction of a 500-million-word reference corpus of contemporary written Dutch, *Essential Speech and Language Technology for Dutch: Results by the STEVIN-programme*, Springer Verlag, chapter 13.

Pauls, A. and D. Klein (2012), Large-scale syntactic language modeling with treelets, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, pp. 959–968.

Sánchez-Cartagena, V.M., F. Sánchez-Martínez, and J.A. Pérez-Ortiz (2011), Enriching a statistical machine translation system trained on small parallel corpora with rule-based bilingual phrases, *Proceedings of Recent Advances in Natural Language Processing*, Hissar, Bulgaria, pp. 90–96.

Schukat-Talamazzini, E.G., R. Hendrych, R. Kompe, and H. Niemann (1995), Permugram language models, *Fourth European Conference on Speech Communication and Technology (EUROSPEECH)*, Madrid, Spain, pp. 1773–1776.

Sidorov, G., F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernndez (2014), Syntactic n-grams as machine learning features for natural language processing, *Expert Systems with Applications* **41**, pp. 853–860.

Stolcke, A. (2002), SRILM an extensible language modeling toolkit, *Proceedings International Conference Spoken Language Processing*, Denver, Colorado, pp. 901–904.

Tyers, F.M. (2009), Rule-based augmentation of training data in Breton-French statistical machine translation, *Proceedings of the 13th Annual Conference of the EAMT*, Barcelona, Spain, pp. 213–217.

Van den Bosch, A., B. Busser, S. Canisius, and W. Daelemans (2007), An efficient memory-based morphosyntactic tagger and parser for dutch, *in* Van Eynde, F., P. Dirix, I. Schuurman, and V. Vandeghinste, editors, *Computational Linguistics in the Netherlands 2006. Selected papers from the seventeenth CLIN Meeting*, pp. 191–206.

Van Noord, G. (2006), At last parsing is now operational, *TALN06. Verbum Ex Machina. Actes de la 13e conférence sur le traitement automatique des langues naturelles* pp. 20–42.

Van Noord, G. (2008), Huge parsed corpora in LASSY, *in* Van Eynde, F., A. Frank, K. De Smedt, and G. Van Noord, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, pp. 115–126.

Vandeghinste, V. (2009), Tree-based target language modeling., *in* Màrquez, L. and H. Somer, editors, *EAMT-2009: Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, Barcelona, Spain, pp. 152–159.

Vergyri, D., K. Kirchhoff, K. Duh, and A. Stolcke (2004), Morphology-based language modeling for Arabic speech recognition, *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH 2004)*, pp. 2245–2248.

Wiktionary (2015). http://en.wiktionary.org/wiki/Category:Dutch_separable_verbs.

Zhang, L. and H. Wang (2014), Go climb a dependency tree and correct the grammatical errors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 266–277.