

Manual sorting of numerals in an inflective language for language modelling

Gregor Donaj · Zdravko Kačič

Received: 10 September 2013 / Accepted: 2 March 2014 / Published online: 17 March 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract In speech recognition systems language models are used to estimate the probabilities of word sequences. In this paper special emphasis is given to numerals—words that express numbers. One reason for this is the fact that in a practical application a falsely recognized numeral can change important content information inside the sentence more than other types of errors. Standard n -gram language models can sometimes assign very different probabilities to different numerals, according to their relative frequencies in training corpus. Based on the assumption that some different numbers are more equally likely to occur, than what a standard n -gram language model estimates, this paper proposes several methods for sorting numerals into classes in an inflective language and language models based on these sorting techniques. We treat these classes as basic vocabulary units for the language model. We also expose the differences between the proposed language models and well known class-based language models. The presented approach is also transferable to other classes of words with similar properties, e.g. proper nouns. Results of experiments show that significant improvements are obtained on numeral-rich domains. Although numerals represent only a small portion of words in the test set, a relative reduction in word error rate of 1.4 % was achieved. Statistical significance tests were performed, which showed that these improvements are statistically significant. We also show that depending on the amount of numerals in a target domain the improvement in performance can grow up to 16 % relative.

Keywords Speech recognition · Language models · Numerals · Manual sorting

1 Introduction

In speech recognition language models are used to score hypotheses generated by the search algorithm (Aubert 2002). The language model estimates the likelihood that a sequence of words can occur in the given language. The most frequently used language models today are standard n -gram language models (Rapp 2008). Those models statistically estimate the probability of a word given $n - 1$ preceding words.

From a language modelling point of view numerals are rather special words. Often a numeral in a sentence can be exchanged with another one and the sentence will still be grammatically correct and both sentences will make sense from a semantic point of view. However a falsely recognized numeral can result in a crucial change of the main message of the sentence, like recognizing the wrong result in news about a sports event. The resulting new sentence will have a different meaning, but on the language level both sentences can not be distinguished by which one should be preferred by a speech recognition system. Even a person looking at both hypotheses may not be able to decide which one could be more probable.

There are of course exceptions like dates and times for example. If a speech recognition system would return the hypothesis “on the 40th of January”, one would suspect a recognition error. Also numerals presenting some small numbers are an exception as they clearly do appear more often in a language and can be found in phrases, e.g. four corners, the seven seas, where we can not assume equal likelihood of different numerals. Probabilities of a given word with its context in a standard n -gram language models are

G. Donaj (✉) · Z. Kačič
Faculty of Electrical Engineering and Computer Science,
University of Maribor, Maribor, Slovenia
e-mail: gregor.donaj@um.si

Z. Kačič
e-mail: zdravko.kacic@um.si

estimated based on frequency counts. Due to the very high number of possible contexts in large vocabulary applications and the limited sizes of training corpora it is very unlikely that two words would have the same probability estimates given the same context. However, there are some words, which seem reasonable to have the same probabilities. For example, let us look at the sentences: “wait 35 minutes” and “wait 45 minutes”. Both sentences are grammatically correct and make the same amount of sense. We could say that both sentences are equally likely to appear in the given language. It is however very unlikely that both numerals will occur exactly the same number of times in the context of the words *wait* and *minutes* in a corpus. Consequently the language model assigns different probabilities to the two sentences. If the corpus is rather small, those differences can become even more evident especially since many different numerals exist.

There are countless numbers. Still, we need only a relatively small amount of different words for writing all possible numerals: one, two, three, etc., 10, 20, 30, etc, 100, 1000, etc. In some languages we need more words since the numbers between 11 and 99 are written as one word. In inflective languages this number increases due to different word forms. As we will show later in this paper the relative frequencies of different numerals can significantly differ due to data sparsity and the limited size of corpora.

The aim of this paper is to propose methods for a manual sorting of numerals in an inflective language into classes, based on the languages characteristics, and to propose a generalization of class-based language models for better performance on speech recognition in a numeral rich domain. Errors with numerals can be more critical to a user who reads hypotheses from a speech recognition system, since the substitution of numerals can change important information in the sentence while other errors can be less critical for correct understanding of the message.

It is our aim in this work to model the probabilities of classes of numerals instead of individual numerals. We therefore present different methods of defining such classes and evaluate the generated language models in large vocabulary continuous speech recognition (LVCSR). Our experiments are performed on a Broadcast News database, which is rather general in speech characteristics. However, one can easily consider to use speech recognition on more special domains like financial and sports news, which contain more numerals.

1.1 Other word classes

The argument presented for numerals can be transferred to other word classes. One example are proper nouns. Like before let us look at two sentences: “I am speaking with James” and “I am speaking with Jethro”. Again, both sentences make equally sense, but their language model score will differ based on the frequencies of these two names.

Since we are only trying to show a possible example we will not look at population statistics and just assume that James is a popular name and Jethro is a rather rare name. Suppose that we try to recognize the utterance “I am speaking with Jethro” and that the search algorithm has considered both of the above sentences as hypotheses. If the pronunciation was clean and no other acoustic disorders were present the hypotheses with “Jethro” should have a higher acoustic score. However, if the rare name occurs in the training corpus rarely enough the difference in language models score could be higher and the speech recognizer will say “James”. A similar example could also be presented for geographical names (e.g. Valencia and Palencia – two cities in Spain) and other proper nouns.

1.2 Previous work

Previous work that was focused on modelling numerals was done mostly in an application specific domain with small vocabularies, e.g. the recognition of digits or natural numbers over telephone lines (Kvale 1996; Kurian and Balakrishnan 2009). In (Ghanty et al. 2010) Bengali numerals were considered in a isolated word recognition task. Sproat (2010) used expansion of numerals in text normalization for text-to-speech synthesis. Taking into account the characteristics of Russian, which are similar to the characteristics of Slovene, generative and discriminative language models were presented.

Other similar work was considered with well known class-based language models (Whittaker and Woodland 2003). Those models sort vocabulary words into classes based on different criteria. Usually all words in the vocabulary are sorted into classes. The models proposed here can also be seen as a special case of generalized class-based language models. However, we found no previous work that specifically concerned numerals in LVCSR in a general domain like Broadcast News.

Proper names are also a topic of interest in pronunciation (Reveil et al. 2012; Schlippe et al. 2014), which is also important for speech recognition.

Huet et al. (2010) proposed a post-processing method of speech recognizer hypotheses, which includes morpho-syntactic description tagging. In the post processing step consecutive numerals and consecutive proper names are grouped into single cardinal and proper names tags. In their method a number, which is written as several words, can be treated as one single token in the morpho-syntactic language model.

1.3 Statistical tests

With numerals we address some of the words that appear in speech recognition. Depending on a given specific domain the rate of numerals can be either small or large. Therefore we

expect a corresponding small or large improvement in performance. Normally large improvements are not questioned whether they are the results of a genuine improvement of the recognition system or the happened by coincidence. To estimate if small improvements are significant we perform statistical significance tests.

Most statistical significance test give us a result in form of a p -value that is between 0 and 1. Without going into to much detail, we can say that improvements are called statistically significant if the p -value lies below the significance level α . Usually we select $\alpha = 0.05$.

The need for statistical tests in speech recognition was early recognized in Gillick and Cox (1989) where test for isolated word recognition were proposed. However, only few papers report significance tests results in addition to bare performance results. A more recent example is in Bisani and Ney (2004) where bootstrap resampling tests for LVCSR were proposed. Another type of statistical test for natural language applications is approximate randomization (AR) (Riezler and Maxwell 2005), widely used in machine translation tasks. AR makes less assumptions about the test set than bootstrap resampling. In Riezler and Maxwell (2005) it is shown that AR test are more conservative, since they usually give larger p -values. We decided to use AR tests to estimate the significance of our results.

1.4 Organization of the paper

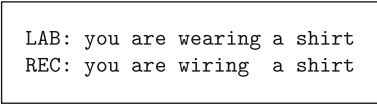
This paper is organized as follows. Section 2 presents a few examples of recognition errors and our motives for modelling numerals. In Sect. 3 we present Slovenian numerals, their writing rules and grammatical rules for matching. In Sect. 4 we describe the proposed sorting and modelling methods as well as their possible application to other languages and other word classes. In Sect. 5 the experimental system used to evaluate the proposed methods is discussed. In Sect. 6 the results of word error rates, a comparison with class-based language models, and a more detailed analysis of our best model are given. The conclusion follows in Sect. 7.

2 Recognition errors and numerals

The three types of errors in a continuous speech recognition system are substitution, deletion and insertion of words. The most widely used metric for evaluation LVCSR systems is word error rate (WER), defined by the equation:

$$E = \frac{S + D + I}{N}, \quad (1)$$

where S , D , I and N are the numbers of substitutions, deletions, insertions and total words respectively. WER is an objective metric that weights all errors equally. However,



LAB: you are wearing a shirt
REC: you are wiring a shirt

Fig. 1 Substitution error in speech recognition

in a practical application different errors may have different impact on the subjective perception of the performance of a LVCSR system. To clarify this thought, let us look at some examples of possible recognition errors. Although this paper describes the recognition of numerals in Slovene, for better understanding these examples are in English.

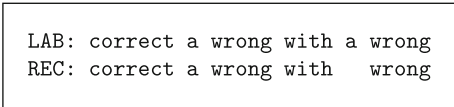
The example in Fig. 1 shows a reference transcription (LAB) and a recognizer hypothesis (REC) with a substitution error. If one reads the hypothesis, this sentence will seem to be nonsense. A reader would probably also realize that the word *wearing* was substituted in the recognition process and therefore might guess the actual spoken sentence.

An example of a deletion error is shown in Fig. 2. A reader would probably also recognize the error in this example and understand the spoken sentence correctly from the recognizer hypothesis.

Figure 3 shows a substitution example, where one numeral was replaced by another one. Looking at both the reference transcription and the recognizer hypothesis it is not possible to tell which one is correct, since both sentences are grammatically correct and both make sense. A reader looking at the hypothesis would think that the sentence is correctly recognized.

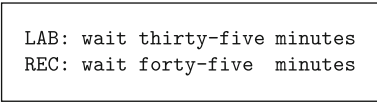
One more frequent type of errors in Slovene and other inflectional languages is the false recognition of word endings. Those endings define the grammatical role of a word inside the sentence. For example a word could be recognized in its plural form instead of singular. In most cases these errors may not hinder one to understand the meaning of the sentence. However, they make the sentence harder to understand.

From those examples we can conclude that substitution errors of numerals can be more critical in a practical applica-



LAB: correct a wrong with a wrong
REC: correct a wrong with wrong

Fig. 2 Deletion error in speech recognition



LAB: wait thirty-five minutes
REC: wait forty-five minutes

Fig. 3 Substitution-of-numerals error in speech recognition

tion than some other types of errors. If numerals in a language are written in several words, as this is the case with cardinal numerals in Slovene, also deletion and insertion errors can have a similar effect.

3 Numerals in the slovenian language

Slovene (Toporišič 2000) distinguishes cardinal (one, two, three ...), ordinal (first, second, third ...), disjunctive (single, double, triple ...), and multiplicative (once, twice, thrice ...) numerals. In our work we primarily consider cardinal numerals, which are the most frequent. We briefly also consider ordinal numerals.

3.1 Inflection

Slovene is an inflectional language. Inflectional words (nouns, adjectives, verbs and some adverbs) are inflected according to their grammatical characteristics. The inflection of Slovenian words is reflected by different endings in word forms. The endings of numerals change by grammatical number, gender, and case.

The numeral *one* varies according to its grammatical gender, case and number. The later can be singular or plural. The numeral *two* exists only in dual form. All other cardinal numerals exist only in plural form. The numerals from 2 to 4 vary in case and in the nominative case also in gender. Table 1 shows a short example of the use of the numeral *three* in nominative case. The cardinal numerals from five on vary only in case and have 4 possible endings: *-ih, -im, -imi, -o (empty ending)*. An example is given in Table 2.

All ordinal numerals vary with gender, case, and number. This gives us a larger set of possible endings. Also because

those numerals are all written in one word this means that there are far more words needed to represent those numbers. These are the two reasons, why there is a much larger number of words needed to represent ordinal numbers, than cardinal numbers.

3.2 Writing rules

In Slovenian language the cardinal numerals from 1 to 99 and the numerals 100, 200 ... 900 are written as one word. Those words do not compound with each other. Also the words thousand, million, billion, etc. are written as isolated words. For example the number 12,375 is written in four words as: *dvanajst tisoč tristo petinsedemdeset*. Writing this numeral in different cases will change the ending only on the last word and all previous words stay in nominative case. Thus, a language model shall assign small probabilities to word sequences of numeral words, which do not have this property.

Considering that most cardinal numerals have 4 different endings, we can estimate that we need about 400 different words to write all numerals from 1 to 99. With additional 36 words (4 different word forms for 100, 200, ... 900) we can write all numerals up to 999. We can also add 4 different word forms for thousand, million, billion, etc. Thus, we need only a total of approximately 450 different isolated words to write every cardinal numeral used in everyday speech.

Cardinal numerals on the other hand are always written as one word. For example the number 12,375th is written: *dvanajsttisočtristopetinsedemdeseti*. Almost all ordinal numbers have 11 different endings. Since those numerals are always written in one word, we would need approximately 11,000 different words to write all ordinal numerals only from 1st to 999th.

3.3 Grammatical matching

One important property of Slovenian language is that adjectives (including numerals) and nouns match in gender, case, and number. A language model, trained on a grammatical correct corpus, would assign small probabilities to pairs of

Table 1 The use of *three* in all three grammatical genders

Gender	Slovene	English
Male	Trije moški	Three men
Female	Tri ženske	Three women
Neutral	Tri dekleta	Three girls

Table 2 The use of *ten* in all six grammatical cases

Case	Slovene	English
Nominative	(Mojih)	deset prijateljev (My) ten friends
Genitive	(Ne vidim)	desetih prijateljev (I don't see) ten friends
Dative	(Pošiljam)	desetim prijateljem (I send to) ten friends
Accusative	(Vidim)	deset prijateljev (I see) ten friends
Locative	(Govorim o)	desetih prijateljih (I talk about) ten friends
Instrumental	(Družim se z)	desetimi prijatelji (I hang out with) ten friends

adjectives and nouns, which do not match and higher probabilities to those that do match.

Numerals from *five* on are however an exception. A noun, that follows a numeral, matches it only in four cases. In nominative and accusative case the following noun is in genitive form. That can also be seen in Table 2.

Still we can see that some combinations of numeral forms and noun forms are not grammatically correct. For example if the numeral has the ending *-imi*, it can be only followed by a noun in instrumental form. Thus the ending of a numeral can give us hints of the possible forms of the next word. Therefore it seems not to be reasonable to group numerals with different grammatical characteristics into the same class.

4 Sorting methods for numerals

4.1 The methods

We have tested different methods for manual sorting of numerals, mostly based on writing rules and numeral word endings and therefore indirectly on their grammatical characteristics. The methods also differ in the number of numerals in a class. All together we tested 6 different models.

- **Baseline model:** We did not use any sorting of numerals. They were treated like any other words. This is a standard word-based n -gram model.
- **Writing rules with 4 classes (WR-4):** We took cardinal numerals and grouped them according to writing rules. We split the numerals in two ways: the nominative case from the other cases and the numerals expressing the numbers from 1 to 99 from the others. Thus, we got 4 classes with 447 numerals in total. This sorting was based on the writing rules and possible sequences of word expressing cardinal numerals.
- **Writing rules with 6 classes (WR-6):** We split the classes obtained by method WR-4 further into classes with numerals expressing numbers from 1 to 10 and classes with numbers 11–99. We got 6 classes with 447 numerals in total. This method was chosen on the assumption that smaller classes give better results.
- **Word endings with 4 classes (END-4):** We included only cardinal numerals expressing the numbers from 11 to 99. Each of these numerals has only 4 possible endings. We separated them into 4 classes according to their endings. The classes included 89 numerals each. This technique was based on the property of word matching in the Slovenian language.
- **Word endings with 12 classes (END-12):** Again we took only cardinal numerals and separated them according to their endings and to the numbers they express. We grouped

numerals expressing numbers from 5 to 10, from 11 to 99 and the numbers 100, 200, ... 900. We got 12 classes with 416 numerals total.

- **Word endings with 12 classes and ordinal numerals (END-12+ORD):** We took the classes from the previous method and added a 13th class with 10.879 ordinal numerals from 11th to 999th. Ordinal numerals from 1st to 10th were not sorted.

4.2 Language model probabilities and relation to class-based models

To each class of numerals a label was assigned. A class acts as a single vocabulary entry when building language models. Thus probabilities are calculated only for classes. For example, take the sentence *I saw ten things*. A classical bigram language model would assign it the probability:

$$\begin{aligned}
 P(\text{I saw ten things}) &= P(\text{I} | \langle s \rangle) \\
 &\cdot P(\text{saw} | \text{I}) \\
 &\cdot P(\text{ten} | \text{saw}) \\
 &\cdot P(\text{things} | \text{ten}) \\
 &\cdot P(\langle /s \rangle | \text{things}), \quad (2)
 \end{aligned}$$

where $\langle s \rangle$ and $\langle /s \rangle$ are start-of-sentence and end-of-sentence markers respectively.

Let us say that in some of the proposed models the numeral *ten* is in a class labeled $\langle \text{number_A} \rangle$. The new models would assign the probability:

$$\begin{aligned}
 P(\text{I saw ten things}) &= P(\text{I saw} \langle \text{number_A} \rangle | \text{things}) \\
 &= P(\text{I} | \langle s \rangle) \\
 &\cdot P(\text{saw} | \text{I}) \\
 &\cdot P(\langle \text{number_A} \rangle | \text{saw}) \\
 &\cdot P(\text{things} | \langle \text{number_A} \rangle) \\
 &\cdot P(\langle /s \rangle | \text{things}), \quad (3)
 \end{aligned}$$

These models are similar to class based models, where the probability of the word *ten* would be calculated as the product

$$\begin{aligned}
 P(\text{ten} | \text{saw}) &= P(\langle \text{number_A} \rangle | \text{saw}) \cdot \\
 &P(\text{ten} | \langle \text{number_A} \rangle), \quad (4)
 \end{aligned}$$

where the first factor is the probability that a word from the class $\langle \text{number_A} \rangle$ follows the word *saw* and the second is the probability that the word *ten* occurs given the occurrence of the class $\langle \text{number_A} \rangle$.

The proposed models do not assign different probabilities to different words in a class. In fact if we would describe

these models as class-based, we would say that each word in a class has a probability of 1.

A generalization of the standard class-based language model, which also includes the proposed model can be presented with the equation

$$P(w_i|w_{i-1}) = P(c_i|w_{i-1}) \cdot P(w_i|c_i)^\alpha. \quad (5)$$

The equation describes the bigram case, but it can be easily adapted for higher order models. The parameter α is a real number. In a standard class-based language model the value of α is 1. In the proposed model α is 0. The word w_i belongs to the class c_i . One could say that we constructed a theoretically faulty class-based language models since the sum over all words is not 1. Our assumption was that such a model is not necessary the best, at least not, if it comes to numerals. However, our models uses numeral classes and other words as basic vocabulary unit. The sum of all probabilities over those units remains 1.

To conclude theoretical aspects of proposed language model in terms of class-based models, our idea is to assign all numerals in the given class a probability of 1, once the probability of the class is given. Hence these probabilities are independent from the class size. The distinction between numerals from the same class is based only on the acoustical model score.

For a comparison we repeated all our experiments with class-based language models. We used the same sorting and corpus processing techniques. The difference was in the final language models. Here we took the class size into account. All words in a class got the same probability depending on the class size.

4.3 Application to other languages

Slovene shares many similarities with other Slavic languages like Croatian, Serbian, Russian, Czech, Slovak, etc. However, those languages do not share the same writing rules for numerals. We give a few examples.

Slovak has similar writing rules as Slovene. Numerals from 11 to 99 are written as one word, e.g. 21 = *dvadsat-jeden*.

Czech is an example of a Slavic language in which numerals like 21, 22 ... 31, 32 ... 99 are written as two separate words, e.g. 21 = *dvacet jedna*. Similar rules are in Ukrainian.

In Croatian numbers from 1 to 20 are written as one word. Numbers from 21 to 99, except 30, 40 etc., can be written either as one word (21 = *dvadesetjedan*) or as three words (*dvadeset i jedan*). This is the same for cardinal and ordinal numerals.

The formation of numerals from 20 to 99 is often very similar. For tens (20, 30 ... 90) The numerals is one single inflected word. For other numbers the first word or first part presents the tens. This is sometimes followed by the word

and. The last part is the inflected form of numerals presenting 1–9. The differences between languages are in the writing of those parts: there are either written as one word or as separate words. The writing rules for some non-slavic languages are also similar, e.g. in English, where the words are connected by a hyphen.

The numerals in the mentioned languages are inflected by gender and case. Like in Slovene, inflections are reflected by changes in the word ending. This is a well known feature of Slavic languages.

As the word formation rules for numerals in other Slavic and some non-Slavic languages are very much similar, the methods based on writing rules can be easily used in other languages, where numerals are written as one word. With a few additions those methods can also be used for languages where numerals are written as separate words.

For example in Croatian, where numbers 21, 22 ... 99 can be written as separate words with the word *and* in the middle, one could sort the words for writing numerals based on the fact that some of them can appear before the word *and* and are not inflected, while the other words can appear after the word *and* and are inflected.

The methods based on word endings are applicable to other languages with similar word inflection rules.

4.4 Application to other word classes

The proposed idea of manual sorting of numerals can be adapted to make methods for the sorting of other word classes.

A speech recognition system may produce errors, when trying to recognize a sentence in which the first and last name of a person are stated and this combination does not occur in the training corpora. With many possible combinations of first and last names, we can assume that many combinations will appear even in a very large corpus.

A language model, which would use the proposed method of sorting with first and last names as their classes, would assign the same or very similar probabilities to all combinations of first and last names. Therefore it would permit the correct recognition of combinations of first and last names, which are not in the corpora, as long as both names individually appear in the recognizer's vocabulary.

As with small numerals presenting small numbers, we also have combinations of names that are more likely to occur, namely the names of well known people.

A similar case are geographical names. Classes could be formed by grouping cities from a certain state then grouping the name of states in the US, Brazil, Germany and provinces in Canada. Next we could form a class from the names of countries, rivers, mountains, etc.

Some common nouns could be also considered for the proposed methods, e.g. trees, fruits, vegetables, etc.

5 Experimental system

To test the proposed models we performed tests on the Slovenian Broadcast News (BN) database (Žgank et al. 2005). The database is divided into three sets: the train set, which was used to train the acoustical models, the development set, which was later used to optimize model weights and word insertion penalty, and the test set, used to evaluate performance.

The used features for the acoustical models were the log-energy and 12 mel-frequency cepstral coefficients with their first and second derivative. Computation of the features was done on 32 ms Hamming windows with 10 ms spacing. The final acoustical models were grapheme based triphone cross-word models, composed of 16 Gaussian mixture densities.

Language models were trained on the FidaPLUS corpus (Arhar and Gorjanc 2007), which is the largest Slovenian corpus available to us. The first step in building the proposed models was to create partial dictionaries with the proposed classes of numerals. After this step we started the processing of the text corpus. Each word was checked if it equals a numeral from one of the defined classes. If so it was replaced by the label of the corresponding class. The dictionary was constructed from the 100.000 most frequent words in the text corpus and the numerals. The processed corpus was used to build bigram and trigram language models. We applied Good-Turing smoothing and Katz back-off.

For a more detailed error analysis we used a freely available part-of-speech tagger called Obeliks. The tagger was developed at the Institute Jožef Stefan. We tagged reference transcriptions of the test set as well as speech recognition outputs. The tagger identifies word class and other morpho-syntactical information of the words in its input sentences. The possible word classes were defined in the JOS project (Erjavec et al. 2010), which are different then in the formal Slovenian Grammar. In the Grammar, numerals are a subclass of Adjectives. In the JOS specification, the numerals are an independent word class. The types of numerals in the

JOS specifications are cardinal, ordinal, special (other) and pronominal numerals (*one, other*).

6 Results

6.1 Comparison of modeling methods

We tested all 6 models on the test set. The word error rates results are given in Table 3. The results of the proposed models are in the 2nd and 3rd column. The results for the corresponding class-based models are in the 4th in 5th column. The best results with bigram models were achieved with the methods END-4 and END-12, where we achieved an improvement of 0.6 % relatively. With the trigram model we achieved the best results with the END-12 method, namely a 1.4 % relative reduction of word error rate.

The methods WR-4 and WR-6 gave the highest WER. A comparison between these two methods and methods END-4 and END-12 confirms our assumption that smaller classes give better results, although the difference is very small.

Methods END-4 and END-12, which are based on endings of numerals, outperform methods WR-4 and WR-6, which are based on writing rules. Both methods END-4 and END-12 gave better results than the baseline model, while methods WR-4 and WR-6 did not outperform the baseline system.

A comparison between the methods END-12 and END-12+ORD shows an increase in word error rate if ordinal numerals are added to the modelling methods. We assume this is a consequence of the large number of numerals in one class. For a useful inclusion of ordinal numerals we would need different criteria based on which we could do further sorting of ordinal numerals.

Table 4 show significance test results performed between the baseline system and all other. With only a few exceptions all differences between results are statistically significant at the $\alpha = 0.05$ significance level. The fact that small differences are significant is not surprising. The differences in the models are minimal for words other than numerals. Therefore most of the recognition results are the same. Most differences occur at numerals.

Table 3 Word Error Rates in %

Method	Proposed models		Class-based models	
	Bigram model	Trigram model	Bigram model	Trigram model
Baseline	31,46	28,40	31,46	28,40
WR-4	31,81	28,70	31,65	29,31
WR-6	31,78	28,63	31,43	29,80
END-4	31,26	28,03	31,23	28,99
END-12	31,26	28,01	31,17	28,96
END-12+ORD	31,52	28,12	31,15	29,24

Table 4 Significance test results: p -values

Method	Proposed models		Class-based models	
	Bigram model	Trigram model	Bigram model	Trigram model
WR-4	0.015	0.039	0.133	< 0.001
WR-6	0.020	0.105	0.851	< 0.001
END-4	0.019	<0.001	0.003	< 0.001
END-12	0.062	0.001	0.001	< 0.001
END-12+ORD	0.668	0.035	< 0.001	< 0.001

6.2 Comparison with class-based language models

A comparison with the results from the proposed models shows an interesting pattern. The comparison of word error rates of the bigram models shows that the class-based language models always outperform the proposed model. On the other hand, a comparison of the results of trigram models shows right the opposite. All of the proposed models gave better results than the corresponding class-based models. We can observe that the proposed modeling technique performs better with higher order models. It is also notable that with class-based language models method END-12 gave the best results. This indicates that the inclusion of larger classes works better with class-based models.

The obtained results indicate the possibility of a generalization of class-based language models, defined by Eq. (5), by allowing different values for the parameter α .

6.3 Detailed analysis

We will present a more detailed analysis on the results obtained with the END-12 method. The test set contains 1898 segments with total 22743 words. We analysed it with a part-of-speech tagger. We identified 486 cardinal, 129 ordinal, 119 pronominal, and 5 other numerals.

We achieved the best results with the method END-12, where only cardinal numerals were modelled. Therefore we made a more detailed analysis on errors of cardinal numerals. In the baseline system 14 cardinal numerals were deleted, 27 inserted, and 123 substituted with other words. The total error count on cardinal numerals in the baseline system is 164 (33.7 % of all numerals). In the END-12 system 14 cardinal numerals were deleted, 10 inserted, and 73 substituted with other words. In this system the total error count on cardinal numerals is 97 (20.0 % of all numerals). This makes relative error rate reduction of 41 % on cardinal numerals.

Further we separated all sentences in the test set into 2 sets. The first set consists of all sentences containing at least one modelled numeral and the second all other sentences. The term modelled numeral refers to a numeral that was in

one of the classes used in method END-12. We checked the word error rates in those separated test sets and compared it with the baseline model. In the set with numerals we got a word error rate reduction from 26.2 to 22.0 % (16 % relative improvement) and in the set without numerals an increase in word error rate from 28.6 to 28.8 % (0.7 % relative worsening). Although these scenarios are not realistic they give us a worst-case/best-case environment of what we can expect from the proposed models on domains with different amounts of numerals.

7 Conclusion and further work

In this paper we presented the language modelling technique for numerals. It is a variant of class-based language models, that model only the classes itself and not words in it. Although the model is rather simple, it gives better recognition results. Our models are based on grammatical knowledge of the Slovenian language. Similar methods could work with other inflectional languages like Czech, Russian or Polish, and other word classes.

The improvement in word error rate was small, but we showed that in domains with more numerals, like financial or sport news broadcast, we could expect much better results. Moreover, while the error rate on cardinal numerals in the baseline system is much higher than the overall WER, the error rate on cardinal numerals in our best performing proposed model is significantly lower. The initially high error rate on cardinal numerals also confirms our arguments that numerals are harder to model than other words.

We also proposed a generalization of the class-based language model. Models, presented in this paper, and the well known class-based models are special cases of it. Further work in this area would include an optimization algorithm for choosing optimal parameters for such models. The results of this work also indicate that α should depend on the language model order.

Acknowledgments This work was partly financially supported by the Slovenian Research Agency ARRS under contract number 1000-10-310131.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Arhar, Š., & Gorjanc, V. (2007). Korpus FidaPLUS: Nova generacija slovenskega referenčnega korpusa. *Jezik in slovnstvo*, 52, 95–110.
- Aubert, X. L. (2002). An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech & Language*, 16, 89–114.
- Bisani, M., & Ney, H. (2004). Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. *ICASSP 2004 Proceedings (I-409-412)*. Montreal, Quebec, Canada: ICASSP.
- Erjavec, T., Fišer, D., Krek, S., & Ledinek, N. (2010). The JOS Linguistically Tagged Corpus of Slovene. *Seventh International Conference on Language Resources and Evaluation Proceedings* (pp. 1806–1809). Valletta, Malta: LREC.
- Ghanty, S. K., Shaikh, S. H., & Chaki, N. (2010). On Recognition of Spoken Bengali Numerals. *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on* (pp. 54–59).
- Gillick, L., & Cox, S. J. (1989). *Some Statistical Issues in the Comparison of Speech Recognition Algorithms, ICASSP 1989 Proceedings* (pp. 532–535). Glasgow, Scotland: ICASSP.
- Huet, S., Gravier, G., & Sebillot, P. (2010). Morpho-syntactic post-processing of N-best lists for improved French automatic speech recognition. *Computer Speech & Language*, 24, 663–684.
- Kvale, K. (1996). Norwegian numerals: a challenge to automatic speech recognition. *ICSLP 1996 Proceedings* (pp. 2028–2031). Philadelphia: ICSLP.
- Kurian, C., & Balakrishnan, K. (2009). Speech recognition of Malayalam numbers. *World Congress on Nature & Biologically Inspired Computing* (pp. 1475–1479). Coimbatore, India: NaBIC.
- Rapp, B. (2008). N-gram language models for Polish language. Basic concepts and applications in automatic speech recognition systems. *International Multiconference on Computer Science and Information Technology, Proceedings* (pp. 321–324). Wisła, Poland.
- Reveil, B., Martens, J.-P., & van den Heuvel, H. (2012). Improving proper name recognition by means of automatically per name recognition by means of learned pronunciation variants. *Speech Communication*, 54, 321–340.
- Riezler, S., & Maxwell III, J.T. (2005). On Some Pitfalls in Automatic Evaluation and Significance testing for MT. *ACL05 Workshop on intrinsic and extrinsic evaluation measures for MT and/or summarization*.
- Schlippe, T., Ochs, S., & Schultz, T. (2014). Web-based tools and methods for rapid pronunciation dictionary creation. *Speech Communication*, 56, 101–118.
- Sproat, R. (2010). Lightly supervised learning of text normalization: Russian number names. *Spoken Language Technology Workshop (SLT)* (pp. 436–441).
- Toporišič, J. (2000). *Slovenska slovnica*. Ljubljana: Založba Obzorja.
- Whittaker, E. W. D., & Woodland, P. C. (2003). Language modeling for Russian and English using words and classes. *Computer Speech & Language*, 17, 87–104.
- Žgank, A., Verdonik, D., Markuš, A.Z., & Kačič, Z. (2005). BNSI Slovenian Broadcast News Database - speech and text corpus. *9th International conference on Speech communication technology proceedings* (pp. 1537–1540). Lisboa, Portugal: INTERSPEECH.