ITC 4/46

Journal of Information Technology and Control
Vol. 46 / No. 4 / 2017
pp. 425-444
DOI 10.5755/j01.itc.46.4.18367
© Kaunas University of Technology

Evaluation of Language Models over Croatian Newspaper Texts

Received 2017/06/12 | Accepted after revision 2017/10/05

http://dx.doi.org/10.5755/j01.itc.46.4.18367

Evaluation of Language Models over Croatian Newspaper Texts

Slobodan Beliga, Ivo Ipšić, Sanda Martinčić-Ipšić

University of Rijeka, Department of Informatics, Radmile Matejčić 2, 51000 Rijeka, Croatia,
e-mail: {sbeliga, ivoi, smarti}@inf.uniri.hr

Corresponding author: sbeliga@inf.uniri.hr

Statistical language modeling involves techniques and procedures that assign probabilities to word sequences or, said in other words, estimate the regularity of the language. This paper presents basic characteristics of statistical language models, reviews their use in the large set of speech and language applications, explains their formal definition and shows different types of language models. A detailed overview of n-gram and class-based models (as well as their combinations) is given chronologically, by type and complexity of models, and in aspect of their use in different NLP applications for different natural languages. The proposed experimental procedure compares three different types of statistical language models: n-gram models based on words, categorical models based on automatically determined categories and categorical models based on POS tags. In the paper, we propose a language model for contemporary Croatian texts, a procedure how to determine the best n-gram and the optimal number of categories, which leads to significant decrease of language model perplexity, estimated from the Croatian News Agency articles (HINA) corpus. Using different language models estimated from the HINA corpus, we show experimentally that models based on categories contribute to a better description of the natural language than those based on words. These findings of the proposed experiment are applicable, except for Croatian, for similar highly inflectional languages with rich morphology and non-mandatory sentence word order.

KEYWORDS: Statistical language model, Natural language regularity, Word-based language model, Category-based language model, Brown algorithm, POS class, N-gram, Perplexity, Croatian corpora.

1. Introduction

A statistical language model (SLM), or just language model (LM) for short, presents an estimate of word sequences probability distribution [28, 72] or in other words, it is a probabilistic mechanism for text generating. It thus assigns to any sequence of words a potentially different probability. For example, for 3 word long sequences, the language model is defined by probabilities:

p(“She loves him”)=0.001

p(“Loves she him”)=0.000001

where it is important to note that a language model can be context dependent. This means that the associated probabilities of certain sequences of words will not be the same for different corpora. Given a sequence of words as an example: *“She loves him”*, will certainly have a lower probability of occurrence than 0.001 in a corpus of medical texts than in the corpus of romantic novels. Given a language model, we can sample word sequences according to a distribution to obtain a text sample. In other words, we may use such a model to generate text. Thus, a language model is also often called a generative model for text [72].

In general, the purpose of language models is the possibility to find a principled way to quantify the uncertainties, associated with the use of a natural language [28, 72]. More specifically, the main task of a statistical model is to estimate regularities in natural language.

SLMs can be described and observed in a number of different directions. Through the applications in different computer science areas, chronologically following the development and complexity of the models themselves, and through the linguistic lens in applications for different natural language tasks.

1.1. A Chronological Overview and SLM Applications

If we observe chronologically, for many years, SLMs were used mainly in the field of speech recognition [25, 27, 63, 70, 71] and are therefore almost an unavoidable part of any system of statistical speech recognition. As an essential procedure in building effective language models for speech processing procedures, smoothing techniques were introduced (shortly thereafter), but their influence on the model quality regarding the relatively simple n-gram models was small [25].

Significant interest for language models, except for speech processing, was expressed much later in different tasks of natural language processing (NLP): spelling correction, part-of-speech (POS) tagging, syntactic parsing, word and sentence segmentation, shallow parsing, etc. [63]. Additionally, thanks to the language models, in the late 1990s a new approach in the field of information retrieval (IR) has been developed, which is fundamentally different from the traditional probabilistic approach and methods used in vector space models – VSM [72]. The goal of an IR system is to rank documents optimally given a query so that relevant documents would be ranked above irrelevant ones [42, 72]. SLMs provide a principled way of modeling various kinds of document retrieval problems.

The importance of the language models can be found in the observations of a very detailed review from 2003 about the challenges in IR and SLM. According to [3] language models are well adopted for different retrieval, search, extraction and classification tasks.

Information retrieval models, which are used as well as classical TF-IDF (term frequency - inverse document frequency) models. Further improvements require many others techniques in addition to language modeling, while language models are the most promising framework for advancing IR to meet challenges in advanced retrieval tasks.

Cross-lingual information retrieval, where queries are not in the same language as the collection being accessed. Resource requirements are the biggest problem in this field, especially for under-resourced languages.

Web search also uses language modeling. However, even today there are open issues related to the structure of the web, searching, and indexing.

User modeling task takes a large space for LMs applications in order to represent the user by the probability distribution of interests (words), actions (information seeking and user behavior) and annotations (judgments).

Text filtering and classification where semi-structured data and novelty detection are certainly the most researched tasks. Naïve Bayes classification method is actually using a unigram LM estimated for each class of texts from the collection. Despite the inaccuracy of a unigram model, the bag-of-words model is actually effective for solving the text classification problem – especially if the number of classes is small. A similar principle has been adopted for document retrieval. In this model, the distribution of the documents is estimated for two classes (relevant and not relevant), the documents are reduced to attributes so that in the simplest case, they indicate the occurrence of certain words and the attributes themselves are independent (as well as Naïve Bayes model for classification). In comparison with the text or document classification, for retrieval, there is exceptionally little training data, and the only evidence that is available for estimating is the query itself. Thus, the model reduces the whole document to the fact that the document is relevant to the query or not.

Information extraction, question answering, multimedia retrieval and a wide range of **summarization** tasks such as content selection, compression of sentences and documents, generation of headlines, etc. successfully applies language modeling. Heart of many information extraction systems is the language model. In such assignments, we should definitely mention the complex models such as hidden Markov model (HMM) or conditional random fields (CRF). These models in different systems achieve performance comparable to rule-based systems, but the increasing amount of available data requires further investment into more sophisticated language modeling.

From the foregoing, it is evident that SLM has attracted the attention of researchers for more than three decades and it has gained increasing attention recently. In the last 16 years, the quality of models has increased fast. Large amounts of text resources have become available online, and consequently, the number of studies in this area grows very fast.

Many variations of the basic language modeling approaches have since then been proposed and studied, and LMs have now been applied to:

1 **simple statistical tasks** like corpus-based vocabulary list development [31], character-based n-gram classifier that identifies loanwords or transliterated foreign words [35] or personal names [37], Twitter corpus statistics [21], which is further successfully enhanced with demographic data,

2 **multiple retrieval tasks** such as different categorization tasks like authorship retrieval [23] or categorization according to standard text categorization (incoming document is assigned to some pre-existing category) [15] which is robust and tolerant to different kinds of textual errors, such as spelling and grammatical errors, and character recognition errors that comes through optical character recognition (OCR) for contrast to other approaches. In addition, the list of applications continues with the categorization task according to language, either for language identification for written documents [5] or web pages [46]. Action classification in action ontology building using robot-specific texts uses a variety of n-grams as features for supervised machine learning in [44], in order to solve the text classification where action categories are treated as classes and appropriate verb context as classification instances.

Furthermore, language modeling has successfully been applied for various tasks in supervised or semi-supervised settings. The best and well-known supervised methods for automatic keyphrase extraction in their architecture often incorporate SLMs [32]. Some of them have been presented at Workshop on Semantic Evaluation 2010 (SemEval-2010) [32], such as: Humble, Esztergom, SEERLAB, KX_FBK, Maui etc. and achieved remarkable results in contrast to other unsupervised methods that doesn't apply any n-gram statistics. Even deeper than that in the last few years new graph-based methods for the keyword and keyphrase extraction tasks are also in the individual segments based on n-gram statistics. Groups of such methods are elaborated in [7, 8]. Automatic document summarization [6, 20], extractive sentence summarization [66], and a variety of other IR and NLP areas incorporate LMs.

Moreover, LMs are often used in many other fields of artificial intelligence, such as machine translation [63], optical character recognition [16] and handwriting recognition [28].

1.2. Overview of Fundamental Statistical Language Modeling Techniques

The simplest language model is obviously unigram – an n-gram of size 1. Nevertheless, it is not sufficient because it makes unrealistic assumptions about word occurrences in a text [72]. More sophisticated language models have thus been developed to address the limitations of unigram models. A bigram language model can capture any potential local dependency between two adjacent words. N-gram language models would capture some limited dependency between words and assume the occurrence of a word that depends on the proceeding n-1 words [72].

In addition, the next version of the language models is those with "triggers" through which remote dependencies can be captured [58]. If we go even one step further towards sophistication, then certainly we must mention models which are defined through a probabilistic context-free grammar [43]. Rules are used to determine syntactic categories (or part-of-speech symbols) of the words. The nature of these rules is that a certain syntactic category can be rewritten as one or more other syntactic categories or words. The possibilities for rewriting depends solely on the category, and not on any surrounding context, so such phrase structure grammars are commonly referred to as context-free grammars [43]. With these rules, we can derive sentences, and the language model is explicitly structured based on the grammar of a language. Discussion about such LMs can be found in [25, 43].

Bigram and trigram language models tend not to improve much over unigrams. Some of the reasons for this are the problem of data sparseness which makes the estimated complex language models inaccurate [72], and from a lens of retrieval, weaker performance of complex language models may be related to nonoptimal weighting of bigrams and trigrams [48].

Research shows that unigram models are insufficient for machine translation [11], or for speech recognition [25] where modeling word order is obviously very important.

However, there are many other NLP tasks where bigrams nor trigrams are not enough. In such cases, class-based language models can be a solution for improvement of unigram language models. Since the first significant model was proposed in 1980, many attempts have been made to improve the state of the art. N-grams are the staple of current speech recognition technology, and all speech recognition products use some form of an n-gram. 2-grams and 3-grams are a common choice in those systems and probability deriving for their use is still a sparse estimation problem, even for very large corpora [58]. Therefore, maximum likelihood estimation of n-gram probabilities from counts is not advisable. Various smoothing techniques have been proposed. Some of them are recursive backing off to lower order n-grams, linear interpolation n-grams of a different order, variable-length n-grams or a lattice approach [16, 25].

Another known way to battle sparseness is the use of vocabulary clustering. The quality of the resulting model depends on clustering procedures. Thus, some studies show that in narrow discourse domains good results are achieved by manual clustering of semantic categories [69], while some other studies describing manual clustering using linguistic categories (e.g. POS) in less constrained domains show that such an approach does not usually improve over the word-based model. If the model is interpolated with its word-based counterpart, iterative clustering using information theoretic criteria applied to large corpora can sometimes reduce perplexity by 10% [58].

1.3. Class-Based Model Applications

Regardless of the specific application, class-based models have been studied independently for years. Their efficiency is measured in terms of perplexity (PPL). It expresses a weighted average of number of choices, that has to be made by a language model, when calculating the probability of a given test set. Higher values mean, that the model does not fit the testing set very much. A lower number means, that the prediction of the testing set is good (perplexity will be discussed in more details in Sect. 5). Class-based models are usually better than the classic word-based models concerning perplexity. However, there are studies which show that their combination achieves even better results.

Improved clustering techniques for class-based SLM are presented in [34]. Conventional maximum-likelihood criterion was modified using a special form of cross-validation, the leaving-one-out technique. Compared to word bigram, model perplexity is reduced by more than 10% using class-models. Further improvements were achieved by a combination of class-based with word-based models and with part-of-speech models (perplexity reduction of 37%). Similar approaches can be found in the [22]. The word clustering function is heuristically designed and takes into account morphological structure of the words. Between 5 proposed class-based models, one of them has lower perplexity than the baseline language model (PPL reduction over 40%). It has been shown also that the process of interpolation of the class-based language model, using word-clustering function model, with the baseline language model has always caused a significant decrease of the perplexity. With different variations of interpolations decrease of PPL for 7.5 to 42.5% is achieved.

Class-based model applications exist in various areas. First should be mentioned speech recognition, where standard word-based n-gram models had its first use. In the speech recognition task, class-based language models are most commonly used in combination with standard word-based n-gram models in some kind of interpolation, as in the previously described cases.

In [29] authors propose different approaches to class-based language modeling used in a continuous speech recognition system. All of them are based on classes. The experiment shows that better performance of the continuous speech recognition system can be achieved introducing segments of words into class-based language model instead of a classical class n-gram model with classes made up of isolated words.

Authors in [59] present an approach for class-based language modeling based on part-of-speech statistics. More precisely, they investigate approaches to generating a class-based language model based on part-of-speech ambiguity classes. Linear interpolation and word-to-class backoff model for combining the class-based and word-based language models were evaluated and both approaches showed some perplexity improvement and significant reductions in word error-rate for the large-vocabulary speech-recognition task.

Interpolation of standard word-based n-gram models and class-based language models shows small but statistically significant improvement in word recognition accuracy over other standard or class-based models [50].

Morphology-based language modeling is investigated in [33] at different stages in a speech recognition system. Class-based and single-stream factored language models using morphological word representations are applied within an N-best list rescoring framework. Recognition results show perplexity and word error rate reductions.

Class-based language modeling is a long-studied and effective approach to overcome sparse data in the context of n-gram models. In [10] systematic comparison of different forms of class-based models and different class LM combination methods in the context of statistical machine translation for morphologically rich languages is presented. In this study, evaluation is conducted in a large-data scenario and statistically significant BLEU[1] improvement is reported for class-based models in the originally proposed Brown's scenario.

Class-based n-gram language difference models are used in [3] for data selection. A simple method for representing text that explicitly encodes differences between two corpora in a domain adaptation or data scenario is presented. Authors used to select data for a machine translation system in contrast with standard n-gram models, and their language difference models lead to improvements of BLEU in both cases – used in isolation, and used in a multimodel translation system. Language models trained with their method have 35% fewer OOV's than the most common approach. Language models also have a lower perplexity on in-domain data than the baselines.

In addition to speech and statistical machine translation, class-based language model approach is applied to IR tasks, such as named entity identification in [65], sentence retrieval in Question Answering Systems in [49], or keyword extraction in [38]. In [65] class-based LM provides a statistical framework for incorporating Chinese word segmentation and named entity identification in a unified way. Evaluation based on a test data shows that proposed model achieves the performance of state-of-the-art named entity identification systems. For sentence retrieval Brown clustering method is applied in class-based models [49] and, results indicate a significant improvement in terms of mean average precision (from 23.62% to 29.91%). Automatic keyword extraction using meeting transcripts is explored with several approaches in [38]. In the TF-IDF weighting framework, authors incorporate part-of-speech information, word clustering, and sentence salient score. Integrating word clustering was done by inducing class-based n-gram language models. Authors show that unsupervised approach for automatic keyword extraction using meeting transcripts based on TF-IDF approach performs reasonably well, on the other hand using additional information from POS tags provide significantly better results in terms of F-measure.

1.4. Language Models in NLP Applications for Different Natural Languages

Language models are widely used in a variety of languages. We will mention some of them. For example, language models are used for generating diacritics for Arabic names [2]. Microblog language identification for five languages Dutch, English, French, German and Spanish was implemented including n-gram approach [14]. Unsupervised method for developing

[1] BLEU (bilingual evaluation understudy) is a method for automatic evaluation of machine translation.

a character-based n-gram classifier that identifies loanwords or transliterated foreign words in the Korean language as well as a pilot model for Japanese is developed [35]. On the other hand, supervised classification task for Urdu text reuse at document level [60], and party group prediction from the Lithuanian parliamentary speeches is also developed [30].

However, class-based language modeling has a successful application in Arabic [33], Spanish [29], and Japanese speech recognition [70] with improving results of classical models based on words. A comparative study in several languages using automatic and manual word-clustering techniques is presented in [40]. For class-based language models where classes are automatically derived, comparative analysis is presented for five languages: French, British English, German, Italian and Spanish. With regard to classes corresponding to part-of-speech, results are presented for British English, French and Italian. Class-based named entity identification task is studied for the Chinese language in [65]. Class-based models are involved in machine translation experiments for French and English in [3] and for Russian in [10].

One of the biggest problems for word-based language models is data sparsity. This problem is even more emphasized in the case of highly inflectional languages with rich morphology and free word order, such as Arabic, Croatian, Czech, Slovak or Russian language. A method for designing language models for Slovak, a highly inflectional language is presented in [22]. This class-based model shows a significant decrease of perplexity.

In practice, POS category-based language models showed slight advantages compared to conventional n-gram models in speech recognition system for German [19]. Hybrid models that combine categorical models based on POS tags with n-gram models of words also show slightly better results than pure n-gram model of words [68]. However, there are studies that show substantial improvements. For example, in [24] was shown that the categorical language models based on POS tags on the LOB[2] (The Lancaster – Oslo/Bergen) corpus for English reduced the perplexity for 20% compared to the linguistic models based on words.

In this paper, special attention will be given to the Croatian language. For Croatian, word-based n-gram models have been proposed. Most of the research applied statistical models in the field of speech, such as in [51] where authors describe different smoothing techniques applied to language models built from the Croatian weather-domain corpus or in [45] language models are used for acoustic modelling for Croatian speech recognition and synthesis. In addition to speech, language models of the Croatian language are used in other NLP and IR tasks such as: collocation extraction for document indexing [56], keyphrase extraction [47], predicting phrase sentiment [9], term extraction and tagging tools [57].

1.5. Problems and Contributions

The major problem of n-gram language models is data sparsity. This means that a training set does not contain enough data to correctly calculate estimates of the probabilities of a word, based on its history using the most common maximum likelihood method [22]. This problem is even bigger in the case of highly inflectional languages, like Czech, Slovak or Croatian [13, 22]. Furthermore, most studied approaches are limited to use in this languages because they are designed for English or other non-inflectional languages. The problem occurs because of different morphology and word order in sentence for different languages. Generally, in order to reduce the sparseness of the training data and improve model generalization, language models that **group words in categories** have been proposed in [12, 27]. By pooling similar words in the same category, model parameters may be estimated more reliably because they retain patterns for each category in contrast to **models which are based only on words**. With such an approach, it is possible to generalize to word sequences that are not present in a training set within the category.

An example of such categorization of words are models based on POS (part-of-speech) tags that indicate the grammatical functions of words or with some semantic labels such as *company*, *name*, *city*, *date*, *price*, etc. [25]. It is known that bigram language models based on POS categories show competitive performance compared to word-based models (with large data sparseness) for English. However, it is

[2] LOB is a million-word collection of present-day British English texts. Like its American counterpart, the Brown Corpus, it contains 500 text samples of approximately 2,000 words distributed over 15 text categories.

also known that category-based language models are slightly worse than ones based on words when the amount of training material increases [53, 54]. Category-based models are more compact than word-based models (and this is the reason why they outperform in small corpora). However, they are not able to fully exploit all the information available in a large corpus, because they are adapted to capture relationships only between particular categories, but not between particular words. This can be avoided if in accordance with the size of corpus number of categories increases, but in this case, models suffer from high complexity of computation. Bigram and trigram language models based on automatically determined categories can be used in combination with word-based n-gram models [55].

As stated previously, the fundamental limitation of word-based n-gram models is its inability to capture dependencies in a range more than n words. In this way, the model loses much linguistic information which is reflected in the writing style or genre of the text. Empirical evidence suggests that a word which has already been seen in a passage is significantly more likely to recur in the near future than would otherwise be expected. A cache component of the language model addresses this by dynamically increasing the probability of words that have been seen in the recent history of the text. In this way, language model adapts to the local characteristics of the training set but still falls short to address relations within different words. Correlated word pairs are most often revealed by measuring their mutual information (or related measure). Finally, in practice commonly used models are:

1. **word-based n-gram language models** (when the training set is large enough) or
2. **category-based models** with optional accessories to cache components (when the training set is small) [36].

In this paper, we explain and demonstrate the benefits of:

1. **category-based statistical language models** with
   - automatically-determined categories using Brown's algorithm and
   - categories based on Croatian POS tagger, in contrast to
2. **standard word-base n-gram statistical language models** – i.e. Bayesian models.

The rest of the work is conceived as follows: in Section 2, formal definition of statistical language models is described, equivalence mappings of the word history that are crucial for category-based language model understanding is explained in Section 3. A formal definition of a category-based model is presented in Section 4. The quality measure of language models is defined in Section 5, then in Section 6 follows the description of methods, experiment and used dataset. The results are shown in Section 7, and the discussion follows in Section 8. At the end of the paper, final conclusions and possible directions for further research are elaborated.

2. Statistical Language Models

Statistical language model estimates the prior probability of a word sequence $P(\boldsymbol{w}(0, K–1))$, where:

$$\boldsymbol{w}(0, K-1) = \omega(0), \omega(1), \dots, \omega(K-1) \tag{1}$$

is the sequence of K words, and every $\omega(i) \in V$ denotes a word from a fixed and known set of words V – in short: vocabulary [27, 61].

Applying the Bayes' rule of conditional probabilities, $P(\boldsymbol{w}(0, K–1))$ can be decomposed as:

$$P(\boldsymbol{w}(0, K-1)) = \prod_{i=0}^{K-1} \hat{P}(\omega(i)|\boldsymbol{w}(0, i-1)) \tag{2}$$

where $P(\omega(i)|(\boldsymbol{w}(0, i–1))$ is the probability that a word $\omega(i)$ will appear before the word sequence $\boldsymbol{w}(0, i–1)$. The previous word sequence $\boldsymbol{w}(0, i–1)$ is often called a history and is denoted succinctly by h_i. The expression (2) states that the probability of a word sequence $\boldsymbol{w}(0, K–1)$ is given by the probability of the first word, times the probability of the second word given that the first word has appeared before, etc., times the probability of appearing the last word of the word sequence given that all of the previous words have appeared. Therefore, the choice of $\omega(i)$ is modeled to depend on the entire past history of the discourse.

Statistical language models will be considered as probabilistic models – using different ways to assign probabilities to word sequences, whether for computing the probability of an entire sentence or for giving a probabilistic prediction of what the next word will be in a sequence. We will use bigram (2-gram) and trigram (3-gram) language models to determine word sequence probability. Bigram models determine the probability of a word given the previous word, while trigram considers previous two words. The simplest way to approximate probability (in equation 2) is to compute co-occurrences of word sequences – the number of times the word sequence $\omega(i–2)\omega(i–1)$ $\omega(i)$ occurs in the corpus of training data divided by the number of times the word sequence $\omega(i–2)\omega(i–1)$ occurs, written as an expression:

$$P(\omega(i)|\boldsymbol{w}(i-2,i-1)) = \frac{c(\omega(i-2)\omega(i-1)\omega(i))}{c(\omega(i-2)\omega(i-1))} \qquad (3)$$

and called the maximum likelihood (ML for short) estimate.

3. Equivalence Mappings of the Word History

Currently, the most popular statistical language models are **word n-grams**, which we have been previously called Naïve Bayes models (due to the Bayes' rule of conditional probabilities). From a given token (observed frequencies) in the training corpus for word n-gram, the conditional probability was estimated. Specifically, the probability of a particular word is calculated using the frequency of the n-tuple, which consists preceding (n – 1) words of the phrase and the word itself. Such models have the advantage that they are fairly easy to implement and can use larger amounts of training data. However, each tuple is considered independently and fail to keep the basic linguistic patterns from text. Due to the insufficient use of information from the corpus, data fragmentation is an inevitable consequence, which ultimately results in weaker generalizations of tuples that do not appear in the learning set, but still may appear in a real text. Moreover, since the number of n-tuples becomes extremely large as n increases, the models are very complex in terms of the number of parameters they employ. Large sizes of the training set (and consequent memory requirements) together with the sparseness are real drawbacks for large n. For that reason, in actual applications n is usually 2, 3 or 4 (2-gram, 3-gram or 4-gram). It is clear that these models cannot keep associations that involve more than this number of words. Despite these restrictions, language models based on words are still the most successful type of language models that are currently in use.

Let us recall the expression (2) which defines $P(w(0, K–1))$. The language model estimates the conditional probabilities:

$$P(\omega(i)|\boldsymbol{w}(0,i-1)). \qquad (4)$$

Moreover, from this time forth we will refer to $\boldsymbol{w}(0, i–1)$ as the history of the word $\omega(i)$. Due to extremely large number of possible different histories, statistics cannot be gathered for each, and the estimation of the conditional probability must be made on the grounds of some border grouping of $\boldsymbol{w}(0, i–1)$.

We define an operator $H(\omega(i))$ which maps the history $\boldsymbol{w}(0, i–1)$ of the word $\omega(i)$ onto one or more distinct **history equivalence classes**. It can be rewritten as $h_j : j \in \{0,1, \dots ,N_H–1\}$, where N_H denotes the number of different equivalence classes found in the training corpus, so that the classification $H(\cdot)$ segments the words of the corpus into N_H subsets referred to collectively as $\mathbb{H}$:

$$\mathbb{H} = \{h_0, h_1, \dots, h_{N_{H-1}}\}, \qquad (5)$$

where the history operator $H(\cdot)$ is *many-to-one* (M-1), meaning that each word history corresponds to exactly one equivalence class, the conditional probabilities from (4) may be estimated by:

$$P(\omega(i)|\boldsymbol{w}(0,i-1)) \approx P(\omega(i)_i|H(\omega(i))). \qquad (6)$$

Jelinek, Mercer, and Roukos in [12] define the history operator $H(\cdot)$ which generally defines as *many-to-many* (M-M) mapping in which case calculation of the probability involves summing over all history equivalence classes that correspond to $\boldsymbol{w}(0, i–1)$:

$$P(\omega(i)|\boldsymbol{w}(0,i-1)) \approx \\ \approx \sum\nolimits_{\forall h: h\in H(\omega(i))} P(\omega(i)|h) \cdot P\big(h\big|\boldsymbol{w}(0,i-1)\big), \qquad (7)$$

where $P(\omega(i))|\boldsymbol{w}(0, i–1)$ gives the probability that the word history $\boldsymbol{w}(0, i–1)$ belongs to the equivalence class h, and:

$$\sum_{j=0}^{N_{H-1}} P\left(h_j \middle| \boldsymbol{w}(0,i-1)\right) = 1, \forall\, \boldsymbol{w}(0,i-1). \tag{8}$$

Furthermore, equation (7) we can reduce to the equation (6) when $\boldsymbol{w}(0, i{-}1)$ is M-1, since then $P(h)|\boldsymbol{w}(0, i{-}1)$ is nonzero for exactly one equivalence class. For examples, bigram language models define:

$$H(\omega(i)) \overset{\text{def}}{=} \boldsymbol{w}(i-1) = \{\omega(i-1)\} \tag{9}$$

and trigram language models define:

$$H(\omega(i)) \overset{\text{def}}{=} \boldsymbol{w}(i-2, i-1) = \{\omega(i-2), \omega(i-1)\}. \tag{10}$$

where w(i–n,i) refers to the sequence of n+1 words $\{\omega(i{-}n), \omega(i{-}n{+}1), ..., \omega(i)\}$. By equations (10) and (11) word histories are mapped onto the equivalence classes.

4. Category-Based Language Models

Besides the standard language models which find patterns between individual words, language models can be designed in a way of relationships detection between groups of words or categories. To avoid calculation of different histories (which may be numerous), models that will replace the words with their word categories were introduced. Such modeling can achieve some of the advantages [52]:

- Category-based models share statistics between words of the same category and are able to generalize to word patterns never encountered in the training corpus. This ability to sensibly process unseen events is termed language model robustness.
- Grouping words into categories can reduce the number of contexts in a model, and thereby reduce the training set sparseness problem.
- The reduction in the number of contexts leads to a more compact model employing fewer parameters and therefore having more modest storage requirements, which may be important from a practical standpoint.

The category is defined as any grouping of words. Let there be N_v such categories denoted as $\mathbb{V} = \{v_0, v_1, \ldots, v_{N_v}\}$. Then we define the operator $V(\cdot)$ that maps each word $\omega_i : i \in \{0,1, ..., N_\omega\}$ to one or more categories $v_j : j \in \{0,1, ..., N_v\}$, i.e.:

$$v_j = V(\omega_i) \quad j \in \{0,1,\ldots,N_v - 1\} \quad \text{and} \quad i \in \{0,1,\ldots,N_\omega - 1\} \tag{11}$$

where v_j is the category to which ω_i is assigned by the operator $V(\cdot)$. When this mapping is **M-1** we will speak of deterministic category membership, while referring to stochastic membership when it is **M-M**.

If we assume that the probability of witnessing a word $\omega(i)$ is completely defined by the knowledge of the category to which it belongs, then we can write it:

$$P\big(\omega(i)|\boldsymbol{w}(0,i-1)\big) \approx P(\omega(i)|v(i)). \tag{12}$$

For stochastic category membership, this allows us to decompose the conditional probability estimates in the following way:

$$P\big(\omega(i)|\boldsymbol{w}(0,i-1)\big) \approx \sum_{\forall v: v \in V(\omega(i))} P(\omega(i)|v(i)) \cdot P\big(v\big|\boldsymbol{w}(0,i-1)\big). \tag{13}$$

Furthermore, classifying the history into equivalence classes can be derived from equation (7) to:

$$P\left(v_j\middle|\boldsymbol{w}(0,i-1)\right) \approx \sum_{\forall h: h \in H(\omega(i))} P\big(v_j\big|h\big) \cdot P\big(h\big|\boldsymbol{w}(0,i-1)\big). \tag{14}$$

Against this background, a natural choice for the history equivalence class mapping is the identity of the most recent $n-1$ categories:

$$H\big(\omega(i)\big) = \{v(i-n+1), v(i-n+2), \ldots, v(i-1)\} \tag{15}$$

from which we obtain category-based n-gram language models. It is important to note that equation (15) represents **M-M** mapping when the operator $V(\cdot)$ is **1-M**.

When the history equivalence class mapping is **M-1**, equation (14) simplifies to:

$$P\big(v(i)|\boldsymbol{w}(0,i-1)\big) \approx P(v_j|H(\omega(i))) \tag{16}$$

so (14) may be written as:

$$P\big(v(i)|\boldsymbol{w}(0,i-1)\big) \approx \sum_{\forall v: v \in V(\omega(i))} P(\omega(i)|V(\omega(i))) \cdot P(V(\omega(i))|H(\omega(i))). \tag{17}$$

Equation (17) has been used in [27] for synonym-based language models. These models are very similar to the part-of-speech approach except that the categories don't need to have strict grammatical definitions. Instead of a core vocabulary $(\mathcal{V}_{core})$, there is a set of words that are assumed to exhibit all significant types of grammatical behavior that may be encountered. List of synonyms $\mathcal{S}_w$ – list of words that display similar grammatical characteristics to ω, is associated with each word ω in $(\mathcal{V}_{core})$. The synonym lists are compiled automatically from the training corpus by identifying the words in the core vocabulary with which the context of the new word agrees best. In the context of (17), $(\mathcal{V}_{core})$ corresponds to the set of categories, and the synonym set $\mathcal{S}_w$ category membership definitions.

Finally, when we restrict this to determine membership, equation (17) simplifies to:

$$P\big(\omega(i)|\boldsymbol{w}(0,i-1)\big) \approx P(\omega(i)|V(\omega(i)))\cdot P(V(\omega(i))|H(\omega(i))). \tag{18}$$

Except this, using the category n-gram of equation (15) with $n = 2$ from (18) we obtain:

$$P\big(\omega(i)|\boldsymbol{w}(0,i-1)\big) \approx P(\omega(i)|V(\omega(i)))\cdot P(V(\omega(i))|V(\omega(i-1))), \tag{19}$$

which is category-based bigram language model. This model can be used in conjunction with automatically-determined category membership.

4.1. Browns' Algorithm

Brown et al. in [12] introduced an algorithm which assigns word types to disjoint clusters, and is often called Brown's algorithm. It remains a common choice when a simple way to automatically obtain word categories is needed. This algorithm represents an agglomerative clustering procedure which induces a mapping from word types to classes. Training set log probability (LL) for a bigram language model can be written as the sum of unigram distribution entropy $H(\omega)$ and the average mutual information between adjacent categories $I_m(v_1, v_2)$:

$$LL = -H(\omega) + I_m(v_1, v_2). \tag{20}$$

Entropy $H(\omega)$ is defined as:

$$H(\omega) = -\sum_{i=0}^{K-1} P(\omega(i))\log_2(P(\omega(i))), \tag{21}$$

The mutual information between two events x_i and x_j is given by:

$$I_m(x_i, x_j) := log\left[\frac{P(x_i,x_j)}{P(x_i)\cdot P(x_j)}\right]. \tag{22}$$

If the events are taken to be adjacently-occurring words, than $P(x_i, x_j)$ is the probability that x_j immediately follows x_i, and $P(x_i)$ and $P(x_j)$ are the unigram distributions of x_i and x_j respectively. We may estimate these probabilities using relative frequency approximations:

$$\begin{aligned} P(x_i, x_j) &\approx \frac{N(x_i, x_j)}{N(\cdot,\cdot)};\\ P(x_i) &\approx \frac{N(x_i)}{N(\cdot)};\\ P(x_j) &\approx \frac{N(x_j)}{N(\cdot)}, \end{aligned} \tag{23}$$

than for a large corpus $N(\cdot,\cdot) \approx N(\cdot) \equiv N$, and mutual information is defined as:

$$I_m(x_i, x_j) := log\left[\frac{N(x_i,x_j)\cdot N}{N(x_i)\cdot N(x_j)}\right]. \tag{24}$$

The algorithm initially assigns each word in the training corpus to its own category, and then at each iteration merges those category pairs (v_i, v_j) which least decrease mutual information $I_m(v_i, v_j)$. This process continues until the desired number of categories has been reached. In this framework, each word may belong to only one category.

In order to simplify, we can say that the clustering algorithm starts with K classes for the K most frequent word types and then proceeds by alternately adding the next most frequent word to the class set and merging the two classes which result in the least decrease of the mutual information between class bigrams. The result is a class hierarchy with word types at the leaves. The overall runtime of the algorithm is $O(K^2W)$ where K is the number of classes and W the number of word types.

4.2. POS-Based Classes

Words may be classified into groups according to their grammatical function (or part-of-speech) within the sentence. Equations (13), (14) and (15) with $i = 2$, and $i = 3$ are used in the construction of a bigram and trigram language models based on part-of-speech word categories, respectively. The word categories and history equivalence classes must be defined before category-based language models can be used. This work employs a POS tagger for the Croatian language developed by Agić et al. [1]. POS tagging (or POS classification) of words in the training corpus is assumed to be known and constitutes a priori grammatical information that will be exploited by the statistical model. Tokens are replaced with corresponding POS classes with the precondition that each word from a training set has a defined class in the POS list – tuples in the form of [*word; POStag*]. Otherwise, it is associated with the unknown class. In addition to POS list, class distribution list contains frequencies for individual class expansion. For example, if a certain class contains 20% of instances of the entire corpus, then that class will be associated with the probability of 0:2. Thus, the class probability is used instead of pure word occurrences. Tokens that do not appear in the training set are weighted with the default value 1 (parameter *addone*), and classified into unknown class.

5. Language Model Perplexity

Perplexity (*PP* for short) is the most common intrinsic evaluation metric for n-gram language models [28]. An intrinsic evaluation metric is one which measures the quality of a model independent of any application. *PP* is often called a measure of the complexity of a language model. It is related to the entropy assessment, which is defined by the expression:

$$H_p(T) = -\frac{1}{W_T}\log_2 p(T) \tag{25}$$

where $H_P(T)$ is cross-entropy of a model $p(T)$ on the dataset T, and W_T is the number of words in corpus T. Entropy is a measure of the average amount of information contained in a set of sequences that a source can produce. Where the source which can produce a wide range of different sequences will have higher entropy in contrast to those with a limited number of produced sequences. Then perplexity $PP_p(T)$ of a model p in the test set T is defined by the equation:

$$PP_p\ (T) = 2^{H_p(T)}, \tag{26}$$

and is interpreted as the weighted averaged branching factor of a language – the number of possible next words that can follow any word. If perplexity is lower the language model is better. Lower perplexity indicates that language model is closer to the real model [16]. In other words, perplexity is a measurement of how well a probability model predicts a sample. Higher values of perplexity mean that the language model does not fit the testing set very much, while lower indicates that model is good at predicting the sample. The perplexity measure was first proposed by Jelinek, Mercer, and Bahl [26].

6. Data, Methods and Experiments

The experimental objective of this study is to investigate statistical language modeling for Croatian which is a highly inflectional language. In such an environment we want to:

1. build categorical language models for Croatian corpus composed of short texts,
2. compare several different approaches for language models construction: **standard word-based models** with **category-based models** on Croatian news articles from the Croatian News Agency in terms of perplexity:
   - set up and compare different settings of bigram **class-based models with automatically-determined classes using Brown's algorithm** and determine the best number of induced classes in terms of perplexity,
   - set up and compare different settings of bigram and trigram **class-based models based on Croatian POS tags** and determine the best configuration of POS classes in terms of perplexity, and
3. finally, conclude which type of model is the best (standard or some type of class-based) and find n which provides maximum **perplexity reduction** for HINA collection, as well.

6.1. Data

For the purposes of the experiment we use the available part of the Croatian news agency collection – (HINA - cro. *Hrvatska Izvještajna Novinska Agencija*)[3], which is composed of news articles written in Croatian contemporary language. HINA operates according to the principles of an independent, impartial and professional newspaper-reporting agency, and shall not be subject to any influences that could compromise the accuracy, objectivity or credibility of the information, nor factually or legally, to come under the ownership or other interest control of some ideological, political or economic groups. Therefore, the style of their writing is very professional (journalistic), objective and concise.

The HINA collection contains 1020 news articles in XML (Extensible Markup Language) documents. We selected 60 topically diverse documents for the experiment [47]. Two basic criteria were used in the selection of the news articles: the minimum and the maximum size of the document. The length of all 60 texts varies from about 60 to 1,500 tokens – 335 on average. In total, the collection of 60 texts contains 20,125 tokens. Training set has 17,366 tokens, and 10 sets for testing contain the remaining 2,759 tokens. The exact number of tokens for each test set and the total number of sentences are given in Table 1.

Table 1
Corpus statistics

set	no.	sentences	words
training	1	918	17366
test	1	12	227
	2	30	754
	3	11	189
	4	7	88
	5	24	427
	6	31	392
	7	15	370
	8	7	73
	9	8	95
	10	9	144
TOTAL	11	1072	20125

In this experiment, 50 news articles constitute a set for language models construction (training set), and the remaining 10 randomly selected texts present 10 different test sets. Selected texts cover different domains: world news, sports, government, politics, lifestyle, black chronicle, culture, ecology, nature and society, and other life topics.

6.2. Language Modeling Tools

The various software packages for statistical language modeling have been in use for many years. One such package – The CMU (Cambridge Statistical Language Modeling) toolkit [17], has been in wide use in the research community and has greatly facilitated the construction of language models for many practitioners, especially in the first version of toolkit – The CMU SLM (Carnegie Mellon Statistical Language Modeling) [63].

However, in this experiment, the SRILM (The SRI Language Modeling Toolkit) is used [63]. SRILM is a collection of C++ libraries, executable programs, helper and wrapper scripts designed to enable the production and experiment implementation of statistical language models for speech recognition and other applications. The tool supports the creation and evaluation of different language model types that are based on N-gram statistics. Further, the tool goes beyond simple LM construction and evaluation, covering mainly LM applications and allows performing various comprehensive functionality of language processing, such as statistical tagging, rescoring tool that applies language model over a sequence of adjoining N-best lists [62], tool to perform word error minimization on N-best lists [64] or construct confusion networks [41], converting LMs to word graphs, etc. Tool development started in 1995 at Johns Hopkins Summer School for language modeling. Over the years SRILM has substantially evolved but basically is designed and implemented in three layers consisting of libraries, executable tools, and numerous helper and wrapper scripts [63]. Besides the aforementioned SRILM, in the experiment, we used a POS tagger for the Croatian language proposed in [1].

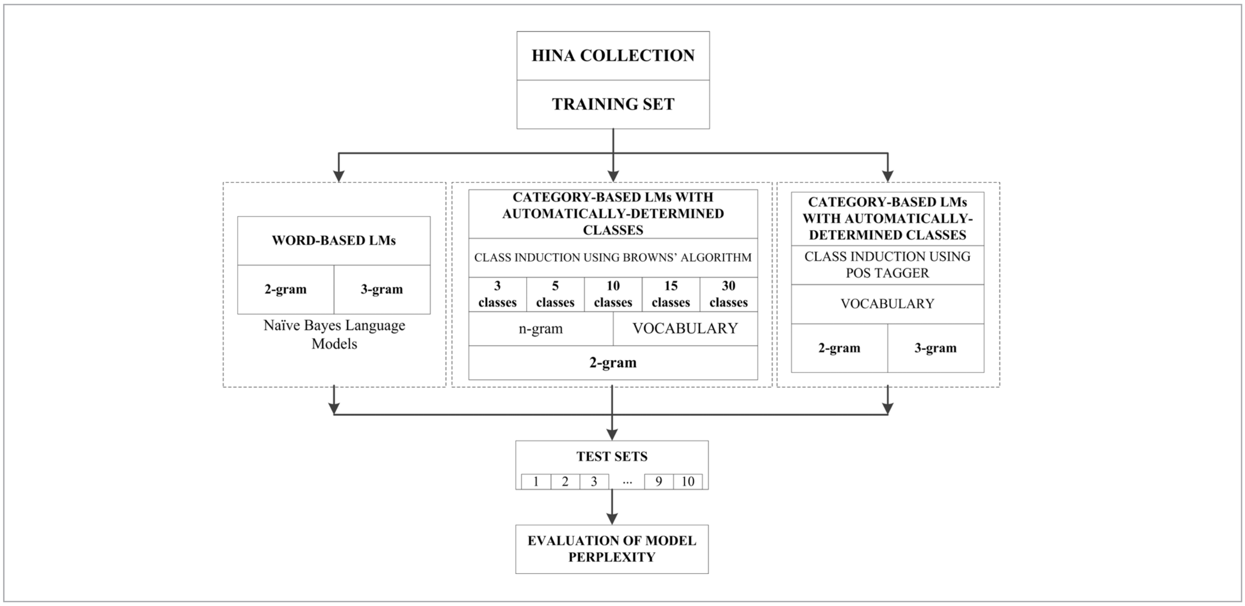
6.3. Experiments

In this work, word sequences will be derived from the Croatian newspaper articles corpus described in subsect. 6.1. In the preprocessing step, texts are separated

[3] https://www.hina.hr/

Figure 1

Schematic diagram of procedures carried out in the experiment


HINA COLLECTION
TRAINING SET
WORD-BASED LMs
2-gram
3-gram
Naïve Bayes Language Models
CATEGORY-BASED LMs WITH AUTOMATICALLY-DETERMINED CLASSES
CLASS INDUCTION USING BROWNS' ALGORITHM
3 classes
5 classes
10 classes
15 classes
30 classes
n-gram
VOCABULARY
2-gram
CATEGORY-BASED LMs WITH AUTOMATICALLY-DETERMINED CLASSES
CLASS INDUCTION USING POS TAGGER
VOCABULARY
2-gram
3-gram
TEST SETS
1 2 3 ... 9 10
EVALUATION OF MODEL PERPLEXITY

from XML tags and cleaned of unnecessary characters and punctuations [.,:;!?/-()]. Three different types of language models are constructed:

- 2-gram and 3-gram language models of words,
- 2-gram category-based language models with 3, 5, 10, 15 and 30 automatically-determined classes using Brown's algorithm,
- 2-gram and 3-gram category-based language models based on POS tags.

Nine different types of statistical language models are constructed in total. Schematic representation of the experiment is presented in Figure 1. Models are built and evaluated according to definitions described in Sect. 2 - Sect. 5.

Category-based language models are based on n-grams respecting belongings of individual words to a particular category. Word classes are induced from word distribution presented in training set – i.e. 2-gram statistics, using **Browns' algorithm** as described in Sect. 4.1. Such a category-based language model with automatically-determined classes using Brown's algorithm is obtained according to the expression (19).

In the first step, category-based language models with POS categories in the first step replace all words from training set with corresponding POS tags (categories). Words (tokens) are replaced with corresponding POS classes with the precondition that each word from the training set has a defined class in the POS list – tuples in the form of [*word; POStag*]. Otherwise, it has been associated with the unknown class. In addition to the POS list, class distribution list contains frequencies for individual class expansions. As defined above, language model building requires a vocabulary of the entire Croatian news agency text collection – HINA. Vocabulary list of HINA collection was developed. It contains entries in tuple form as [*word; POStag*], where in Croatian every word has one of the following POS tags: A - Adjective, V - Verb, N - Noun, M - Numeral, P - Pronoun, S - Adposition, C - Conjunction, Q - Particle, R - Adverb, I - Interjection, Y – Abbreviation, and X - Residual (undefined) according to the MULTEXT-East Morphosyntactic Specifications for the Croatian language [18]. Thus POS tag has the function of a class during the construction of categorical models based on POS tags. With distinction that in a revised version of the MULTEXT-East ver. 4 in [39] abbreviations are assigned as (Y) in our approach, they are merged with residuals and marked as (X).

In the second step, we built the model using the vocabulary and sequences of replaced words with classes. Since there are far fewer POS tags than there are words in a typical vocabulary, the number of different n-grams is much smaller for a given value of n than for a word-based n-gram model. This reduces the problem of data sparseness. Finally, models perplexity is measured in all cases using equation (26).

7. Results

With class-based language models (categorical language models), perplexity can be reduced, as presented by the experimental results. In Figure 2 **bigram language models** of words show lower perplexity in 7 of 10 test cases in contrast to **trigram models** that show lower perplexity in only 3 test sets.

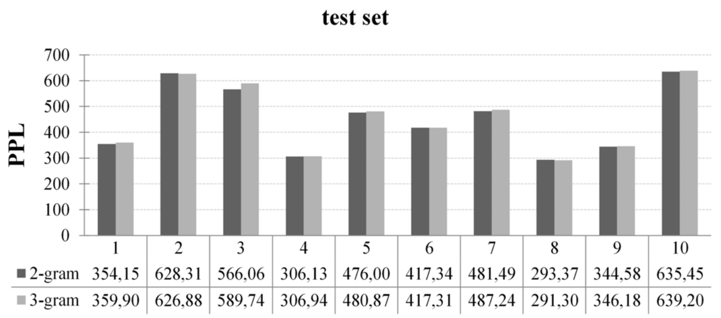
Categorical language model achieves significantly lower perplexity on the individual test sets for a specific number of induced classes. Induction of only 3 classes doesn't achieve better results than those measured for a bigram or trigram language models of words. Due to the limited number of classes language model is unable to make a capable discrimination between different words in the text and the model is too general – overgeneralization. When the number of induced classes increases to 5, categorical language model in 7 of 10 test cases have lower perplexity than bigram models of words. If the number of induced classes is higher (10 and 15) perplexity is significantly reduced. It was expected that the best perplexity will be achieved when the number of induced classes will be equal to 10, because in the Croatian language grammar there are exactly 10 different types of words. However, results obtained with 15 induced classes achieve better perplexity. The results shown in Figure 2 confirmed that perplexity on 10 or sometimes 15 induced classes are significantly better than those achieved on bigram and trigram language models based on words. Thus, increasing the number of classes, the scope of which model should generalize decreases, while the ability of model discrimination is improved – tradeoff. When the number of classes significantly increases (30) the language model begins to reflect the peculiarities of the training set, and generalizes less well to the test set – overfitting (see Figure 3).

Language models based on a POS tagger (with 10 classes), where classes are defined by the types of Croatian words (5 changeable and 5 unchangeable), provide the lowest perplexity – see Figure 4. The bigram POS models are insignificantly better than trigram (Figure 5), probably due to the relatively small corpus. This is possible indication that language models are sensible to the size of the modeled corpus – i.e. trigram models can outperform models based on POS tags if the corpus is larger.

To summarize the main points of the results, in this experiment on the HINA collection of newspaper articles, categorical language models based on POS tags indicate the best results on average in contrast to categorical models with automatically-determined classes and bigram or trigram language models.

Figure 2

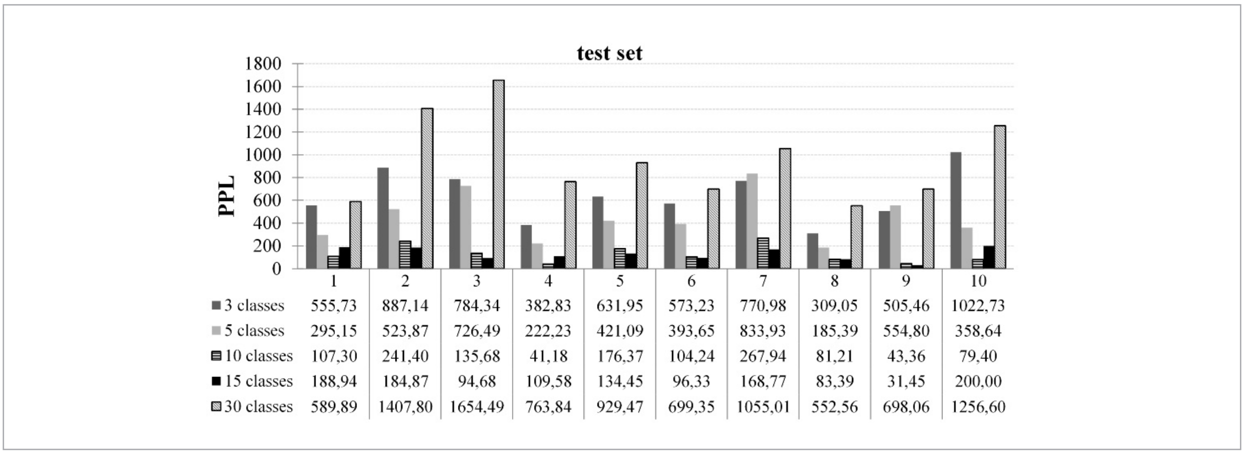
Comparison of perplexity for 2-gram and 3-gram word-based language models on 10 test sets



	1	2	3	4	5	6	7	8	9	10
■ 2-gram	354,15	628,31	566,06	306,13	476,00	417,34	481,49	293,37	344,58	635,45
■ 3-gram	359,90	626,88	589,74	306,94	480,87	417,31	487,24	291,30	346,18	639,20

Figure 3

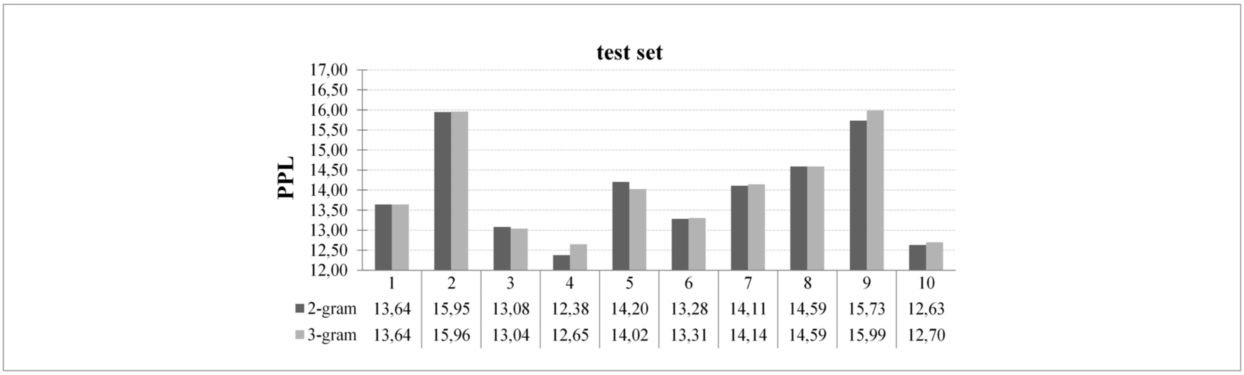
Comparison of perplexity for 2-gram category-based LM with automatically-determined classes (3, 5, 10, 15 and 30) by Browns' algorithm on 10 test sets



	1	2	3	4	5	6	7	8	9	10
■ 3 classes	555,73	887,14	784,34	382,83	631,95	573,23	770,98	309,05	505,46	1022,73
■ 5 classes	295,15	523,87	726,49	222,23	421,09	393,65	833,93	185,39	554,80	358,64
▤ 10 classes	107,30	241,40	135,68	41,18	176,37	104,24	267,94	81,21	43,36	79,40
■ 15 classes	188,94	184,87	94,68	109,58	134,45	96,33	168,77	83,39	31,45	200,00
▨ 30 classes	589,89	1407,80	1654,49	763,84	929,47	699,35	1055,01	552,56	698,06	1256,60

Figure 4

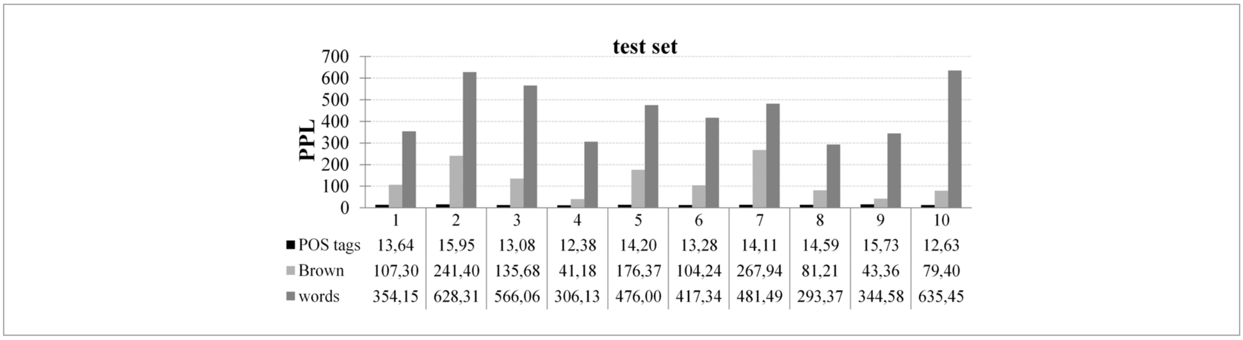
Comparison of perplexity for 2-gram and 3-gram category-based LM with automatically-determined classes by POS categories on 10 test sets



	1	2	3	4	5	6	7	8	9	10
■ 2-gram	13,64	15,95	13,08	12,38	14,20	13,28	14,11	14,59	15,73	12,63
■ 3-gram	13,64	15,96	13,04	12,65	14,02	13,31	14,14	14,59	15,99	12,70

Figure 5

Comparison of perplexity for 2-gram word-based LM with a category-based LM with 10 automatically-determined classes by Browns' algorithm and with automatically-determined classes by POS tags on 10 test sets, respectively



	1	2	3	4	5	6	7	8	9	10
■ POS tags	13,64	15,95	13,08	12,38	14,20	13,28	14,11	14,59	15,73	12,63
■ Brown	107,30	241,40	135,68	41,18	176,37	104,24	267,94	81,21	43,36	79,40
■ words	354,15	628,31	566,06	306,13	476,00	417,34	481,49	293,37	344,58	635,45

8. Discussion

The underlying assumption of an experiment conducted in this work is the possibility of grouping language patterns into syntax groups (structure imposed by grammatical rules) – i.e. using Brown's algorithm or in another solution POS tagger. Categorical language models show the ability to reduce data fragmentation which is inherent in word n-grams. It was expected that syntactic patterns will be more consistent than trivial word occurrence (n-tuples), and thus, produce better generalization to text styles and topics different from those of the training corpus.

The results of the experiment confirmed that the word order has an important role for grammar correctness and that the same was retained by n-grams naturally. A priori information about the grammatical significance obtained under the POS classification is sufficient to describe the syntactic categorical language model.

There are still natural languages that do not have developed complex tools for NLP or they are only partially developed. These are usually those who have a small number of native speakers, and commercial needs for development are not in their focus. Especially considering the complexity of the language and the effort required for development. Therefore, using the POS tagger is not always possible. In this case, human experts perform the tagging manually. Such work is highly impractical and expensive for large amounts of text. Linguistic parsing techniques based on rules may be considered for use in this case. Also, parsing can be computationally very demanding. However, this can be circumvented by tagging only the initial part of the training corpus, which is subsequently used to initialize a statistical tagger for tagging the rest of the text with the most likely category tags. When that's not possible, categorical language models with automatically-determined-categories (with class induction by Brown algorithm, or some other) can be applicable, as shown in this paper.

Results of our work concur with that presented in [58], and carries a slightly larger percentage of perplexity reduction although the HINA corpus is considerably smaller. For all 10 test sets, category-based language models with 10 automatically-determined classes reduce perplexity about 28% on average (average perplexity is 127.81), during category-based models based on POS tags reduces perplexity for 31% on average (average perplexity is 13.96). We can conclude that for Croatian corpus (language from the Slavic group) is worth noticing the fact that perplexity of category-based language models can be reduced, as opposed to word-based models, especially POS-based categorical models. Similar studies of the Slovak language (also from the Slavic group of languages) [27], and for Lithuanian (from the Baltic group of Languages) [67] confirms the same.

9. Conclusion

N-gram model of words is currently the most popular statistical language model which estimates the probability of the observed n-tuples from a training set. This paper presents a comparison of n-gram models based on words with categorical language models based on automatically-determined categories using Brown's algorithm, and categorical language models based on categories determined by a POS tagger. Models are built for the Croatian newspaper articles collection – HINA.

Experimental results are expressed in terms of perplexity – a measure that allows an independent assessment of the language model quality. In this study, a fundamental limitation of the n-gram approach was determined: it is not possible to keep the dependence range of more than n words within the n-gram models based on words. Therefore, the model is not able to address longer-range word-pair relationships that arise due to factors such as the topic or the style of the text. Category-based models with automatically-determined classes demonstrate the ability to circumvent the problem of data sparsity and provide the lower complexity of the n-gram models based on words, especially when the numbers of induced classes are 10 or 15.

Category-based model based on POS tags, in certain configurations, expose improvements in contrast to conventional word-based models. On the HINA collection, they are better than models based on words for 31% on average in terms of perplexity. This confirms that the POS tagger can collect sequential grammatical dependencies from the corpus. Moreover, the model successfully assigns words to POS classes and therefore makes more reasonable predictions for histories that we have not been previously seen and assumes that they are similar to other histories that we have seen before.

As shown in similar studies, empirical evidence suggests that a word which has already been seen in a passage is significantly more likely to recur in the near future than would otherwise be expected. Hence, in future work, it is possible to build a hybrid model that is an approximation of category-based model and classical model based on words. It is also possible to examine the complexity of the model with the addition of a cache language model component, which considers the history of the words in a text.

Acknowledgments

This work has been supported in part by the University of Rijeka under the Language Networks (13.13.2.2.07) and Natural and Multimodal Man-Machine Communication (13.13.1.3.04) projects.

References

1. Agić, Ž., Ljubešić, N., Merkler, D. Lemmatization and Morphosyntactic Tagging of Croatian and Serbian. Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing (BSNLP 2013), Sofia, Bulgaria, Association for Computational Linguistics, 2013, 48-57.
2. Al-Anzi, F. S. Stochastic Models for Automatic Diacritics Generation of Arabic Names. Computers and the Humanities, 2004, 38(4), 469-481. https://doi.org/10.1007/s10579-004-2323-6
3. Allen, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, W. B., Dumais, S., Fuhr, N., Harman, D., Harper, D. J., Hiemstra, D., Hormann, E., Hovy, E., Kraaij. W., Lafferty, J., Lavrenko, V., Lewis, D., Liddy, L., Manmatha, R., McCallum, A., Ponte, J., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Singhal, A., Smeaton, A., Turtle, H., Voorhes, E., Weischedel, R., Xu, J., Zhai, C.-X. Challenges in Information Retrieval and Language Modeling. ACM Press, SIGIR Forum, 2003, 37(1), 31-47.
4. Axelrod, A., Vyas, Y., Martindale, M., Carpuat, M. Class-Based N-Gram Language Difference Models for Data Selection. Proceedings of the IWSLT 2015, Da Nang, Vietnam, 2015, 180-187.
5. Baldwin, T., Lui, M. Language Identification: The Long and the Short of the Matter. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, 229-237.
6. Banko, M., Vanderwende, L. Using N-grams to Understand the Nature of Summaries. Proceedings of HLT-NAACL 2004: Short Papers, ACL, Boston, Massachusetts, 2004, 1-4.
7. Beliga, S., Meštrović, A., Martinčić-Ipšić, S. An Overview of Graph-Based Keyword Extraction Methods and Approaches. Journal of Information and Organizational Sciences, 2015, 39(1), 1-20.
8. Beliga, S., Meštrović, A., Martinčić-Ipšić, S. Selectivity-Based Keyword Extraction Method. International Journal on Semantic Web and Information Systems (IJSWIS), 2016, 12(3), 1-26. https://doi.org/10.4018/IJSWIS.2016070101
9. Biđin, S., Šnajder, J., Glavaš, G. Predicting Croatian Phrase Sentiment Using a Deep Matrix-Vector Model. Proceedings of the 9th Language Technologies Conference, Information Society (IS-JT 2014), Ljubljana, Slovenia, 2014, 95-98.
10. Bisazza, A., Monz, C. Class-Based Language Modeling for Translating into Morphologically Rich Languages. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 2014, 1918-1927.
11. Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., Roossin, P. S. A Statistical Approach to Machine Translation. Computational Linguistics, 1990, 16(2), 79-85.
12. Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., Mercer, R. L. Class-Based N-Gram Models of Natural Language. Computational Linguistics, 1992, 18, 467-479.
13. Brychcín, T., Konopík, M. Morphological Based Language Models for Inflectional Languages. Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Prague, 2011, 560-563. https://doi.org/10.1109/IDAACS.2011.6072829
14. Carter, S., Weerkamp, W., Tsagkias, M. Microblog Language Identification: Overcoming the Limitations of Short, Unedited and Idiomatic Text. Lang Resources and Evaluation, 2013, 47(1), 195-215. https://doi.org/10.1007/s10579-012-9195-y
15. Cavnar, W. B., Trenkle, J. M. N-Gram-Based Text Categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, 1994, 161-175.

16. Chen, S. F., Goodman, J. An Empirical Study of Smoothing Techniques for Language Modeling. Computer Speech & Language, 1999, 13(4), 359-394. https://doi.org/10.1006/csla.1999.0128

17. Clakson, P., Rosenfeld, R. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In G. Kokkinakis, and E. Dermatas (Eds.), Proc. EUROSPEECH, Rhodes, Greece, 1997, 1, 2707-2710.

18. Erjavec, T., Krstev, C., Petkevič, V., Simov, K., Vitas, D. The MULTEXTEast Morphosyntactic Specifications for Slavic Languages. Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages, ACL, Budapest, Hungary, 2003, 25-32.

19. Geutner, P. Fuzzy Class Rescoring: A Part-of-Speech Language Model. EUROSPEECH, 1997.

20. Giannakopoulos, G., Karkaletsis, V., Vouros, G., Stamatopoulos, P. Summarization System Evaluation Revisited: N-Gram Graphs. ACM Transactions on Speech and Language Processing (TSLP), 2008, 5(3), 1-39. https://doi.org/10.1145/1410358.1410359

21. Herdağdelen, A. Twitter N-Gram Corpus with Demographic Metadata. Language Resources and Evaluation, 2013, 47(4), 1127-1147. https://doi.org/10.1007/s10579-013-9227-2

22. Hládek, D., Staš, J., Juhár, J. Word Clustering for a Slovak Class-Based Language Model. Journal of Electrical and Electronics Engineering, 2012, 5(1), 85-88.

23. Houvardas, J., Stamatatos, E. N-gram Feature Selection for Authorship Identification. Proceedings of 12th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2006), Varna, Bulgaria, 2006, 77-86. https://doi.org/10.1007/11861461_10

24. Jardino, M. A Class Bigram Model for Very Large Corpus. Proceedings of the International Conference on Spoken Language Processing, Yokohama, Japan, 1994, 2, 867-870.

25. Jelinek, F. Statistical Methods for Speech Recognition. Cambridge, MA, the MIT Press, 1998.

26. Jelinek, F., Mercer, R. L., Bahl, L. R. A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(2), 1983, 179-190. https://doi.org/10.1109/TPAMI.1983.4767370

27. Jelinek, F., Mercer, R. L., Roukos, S. Principles of Lexical Language Modeling for Speech Recognition. S. Furui & M. M. Sondhi (Eds.), Advances in Speech Signal Processing. New York, Marcel Dekker, 651-699, 1991.

28. Jurafsky, D., Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing. Computational Linguistics and Speech Recognition. Cambridge, MA, the MIT Press, 1999.

29. Justo, R., Torres, M. I. Different Approaches to Class-Based Language Models Using Word Segments. Computer Recognition Systems 2, ASC 45, Springer, Berlin, Heidelberg, 2007, 421-428. https://doi.org/10.1007/978-3-540-75175-5_53

30. Kapočiūtė–Dzikienė, J., Krupavičius, A. Predicting Party Group from the Lithuanian Parliamentary Speeches. Information Technology and Control, 2014, 43(3), 321-332. https://doi.org/10.5755/j01.itc.43.3.5871

31. Kilgarriff, A., Charalabopoulou, F., Gavrilidou, M., Johannessen, J. B., Khalil, S., Kokkinakis, S. J., Lew, R., Sharoff, S., Vadlapudi, R., Volodina, E. Corpus-Based Vocabulary Lists for Language Learners for Nine Languages. Lang Resources and Evaluation, 2014, 48(1), 121-163. https://doi.org/10.1007/s10579-013-9251-2

32. Kim, S. N., Medelyan, O., Kan, M.-Y., Baldwin, T. Automatic Keyphrase Extraction from Scientific Articles. Lang Resources and Evaluation, 2013, 47(3), 723-742. https://doi.org/10.1007/s10579-012-9210-3

33. Kirchhoffa, K., Vergyrib, D., Bilmes, J., Duh, K., Stolcke, A. Morphology-Based Language Modeling for Conversational Arabic Speech Recognition. Computer Speech & Language, 2006, 20(4), 589-608. https://doi.org/10.1016/j.csl.2005.10.001

34. Kneser, R., Ney, H. Improved clustering techniques for class-based statistical language modelling. Proceedings of the 3rd European Conference on Speech Communication and Technology, EUROSPEECH'93, Berlin, Germany, 1993, 973-976.

35. Koo, H. An Unsupervised Method for Identifying Loanwords in Korean. Language Resources and Evaluation, 2015, 49(2), 355-373. https://doi.org/10.1007/s10579-015-9296-5

36. Kuhn, R., deMori, R. A Cache-Based Natural Language Model for Speech Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(6), 570-583. https://doi.org/10.1109/34.56193

37. Li, H., Sim, K. C., Kuo, J.-S., Dong, M. Semantic Transliteration of Personal Names. Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007), 2007, 120-127.

38. Liu,F.,Pennell,D.,Liu,F.,Liu,Y.UnsupervisedApproaches for Automatic Keyword Extraction Using Meeting Transcripts. Proceedings of NAACL'09, ACL, 2009, 620-628. https://doi.org/10.3115/1620754.1620845

39. Ljubešić, N. MULTEXT-East Morphosyntactic Specifications, Revised Version 4, 3.8. Croatian Specifications. Accessed on http://nlp.ffzg.hr/data/tagging/msd-hr.html

40. Maltese, G., Bravetti, P., Crépy, H., Grainger, B. J., Herzog, M., Palou, F. Combining Word- and Class-Based Language Models: A Comparative Study in Several Languag-

es Using Automatic and Manual Word-Clustering Techniques. Eurospeech 2001 – Scandinavia, 2001, 21-24.

41. Mangu, L, Brill, E., Stolcke, A. Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks. Computer Speech and Language, 2000, 14, 373-400. https://doi.org/10.1006/csla.2000.0152

42. Manning, C. D., Raghavan, P., Schütze, H. An Introduction to Information Retrieval. Cambridge University Press, 2009.

43. Manning, C. D., Schütze, H. Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, MA, 1999.

44. Markievicz, I., Kapočiūtė-Dzikienė, J., Tamošiūnaitė, M., Vitkutė-Adžgauskienė, D. Action Classification in Action Ontology Building Using Robot-Specific Texts. Information Technology and Control, 2015, 44(2), 155-164. https://doi.org/10.5755/j01.itc.44.2.7322

45. Martinčić-Ipšić, S., Ribarić, S., Ipšić, I. Acoustic Modelling for Croatian Speech Recognition and Synthesis. Informatica, 2008, 19(2), 227-254.

46. Martins, B., Silva, M. J. Language Identification in Web Pages. Proceedings of the 2005 ACM Symposium on Applied computing (SAC ‘05), Lorie M. Liebrock (Ed.). ACM, New York, NY, USA, 764-768, 2005. https://doi.org/10.1145/1066677.1066852

47. Mijić, J., Šnajder, J., Dalbelo Bašić, B. Robust Keyphrase Extraction for a Large-Scale Croatian News Production System. Proceedings of the 7th International Conference on Formal Approaches to South Slavic and Balkan Languages, Zagreb, Croatian Language Technologies Society, 2010, 59-66

48. Mitra, M., Buckley, C., Singhal, A., Cardie, C. An Analysis of Statistical and Syntactic Phrases. Proceedings of RIAO, 1997, 200-214

49. Momtazi, S., Klakow, D. A Word Clustering Approach for Language Model-Based Sentence Retrieval in Question Answering Systems. Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009, 1911-1914. https://doi.org/10.1145/1645953.1646263

50. Moore, G., Young, S. Class-Based Language Model Adaptation Using Mixtures of Word-Class Weights. Proceedings of ICSLP, 2000, 512-515.

51. Načinović, L., Martinčić-Ipšić, S., Ipšić, I. Statistical Language Models for Croatian Weather-Domain Corpus. INFuture2009: Digital Resources and Knowledge Sharing, Zagreb, Croatia, 2009, 333-340.

52. Niesler, T. Category-Based Statistical Language Models. Doctoral thesis, St. John’s College, 1997.

53. Niesler, T. R, Woodland, P. C. Comparative Evaluation of Word- and Category-Based Language Models. Technical Report CUED/F-INFENG/TR.265, Department of Engineering,University of Cambridge, U.K., 1996.

54. Niesler, T. R, Woodland, P. C. Word-to-Category Backoff Languge Models. Technical Report CUED/F-INFENG/TR.258, Department of Engineering, University of Cambridge, U.K., 1996.

55. Niesler, T. R., Woodland, P. C. Combination of Word-Based and Category-Based Language Models. Proceedings of the 4th International Conference on Spoken Language Processing, Philadelphia, U.S.A., 1996, 1, 220-223. https://doi.org/10.1109/ICSLP.1996.607081

56. Petrović, S., Šnajder, J., Dalbelo Bašić, B., Kolar, M. Comparison of Collocation Extraction Measures for Document Indexing. Journal of Computing and Information Technology, 2006, 14 (4), 321-327. https://doi.org/10.2498/cit.2006.04.08

57. Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., Gornostay, T. Term Extraction, Tagging, and Mapping Tools for Under- Resourced Languages. Proceedings of the Terminology and Knowledge Engineering (TKE2012) Conference, Madrid, Spain, 2012, 193-208.

58. Rosenfeld, R. Two Decades of Statistical Language Modeling: Where Do We Go from Here? Proceedings of the IEEE, 2000, 88(1), 1270-1278. https://doi.org/10.1109/5.880083

59. Samuelsson, C., Reichl, W. A Class-Based Language Model for Large-Vocabulary Speech Recognition Extracted from Part-of-Speech Statistics. IEEE Proceedings of Acoustics, Speech, and Signal Processing, 1999, 1, 537-540. https://doi.org/10.1109/ICASSP.1999.758181

60. Sharjeel, M., Nawab, R. M. A., Rayson, P. COUNTER: Corpus of Urdu News Text Reuse. Language Resources and Evaluation, 2016, 1-27. https://doi.org/10.1007/s10579-016-9367-2

61. Song, F., Croft, W. B. A General Language Model for Information Retrieval. Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, U.S.A., 1999, 279-280. https://doi.org/10.1145/319950.320022

62. Stolcke, A. Modeling Linguistic Segment and Turn Boundaries for N-Best Rescoring of Spontaneous Speech. In G. Kokkinakis, N. Fakotakis, E. Dermatas (Eds.), Proceedings EUROSPEECH, Rhodes, Greece, 1997, 5, 2779-2782.

63. Stolcke, A. SRILM – An Extensible Language Modeling Toolkit. INTERSPEECH, 2002, 257-286.

64. Stolcke, A., Konig, Y., Weintraub, M. Explicit Word Error Minimization in Nbest List Rescoring. In G. Kokkinakis, N. Fakotakis, E. Dermatas, (Eds.), Proceedings EUROSPEECH, Rhodes, Greece, 1997, 1, 163-166.

65. Sun, J., Gao, J., Zhang, L., Zhou, M., Huang, C. Chinese Named Entity Identification Using Class-Based Lan-

guage Model. Proceedings of the 19th International Conference on Computational Linguistics, ACL, 2002, 1, 1-7. https://doi.org/10.3115/1072228.1072240

66. Turan, M., Sönmez, C., Ganiz, M. C. The Benchmark of Paragraph and Sentence Extraction Summaries Using Outlier Document Filtering Based Multi-Document Summarizer. Information Technology and Control, 2014, 43(4), 433-439. https://doi.org/10.5755/j01.itc.43.4.7010

67. Vaičiūnas, A., Kaminskas, V., Raškinis, G. Statistical Language Models of Lithuanian Based on Word Clustering and Morphological Decomposition. INFORMATICA, 2004, 15(4), 565-580.

68. Wakita, Y., Kawai, J., Iida, I. An Evaluation of Statistical Language Modeling for Speech Recognition Using a Mixed Category of Both Words and Parts-of-Speech. Proceedings of 4th International Conference on Spoken Language, ICSLP '96, 1996, 1, 530-533. https://doi.org/10.1109/ICSLP.1996.607171

69. Ward, W. H. The CMU Air Travel Information Service: Understanding Spontaneous Speech. Proceedings of the Workshop on Speech and Natural Language, ACL, Stroudsburg, PA, USA, 1990, 127-129.

70. Yokoyama, T., Shinozaki, T., Iwano, K., Furui, S. Unsupervised Class-Based Language Model Adaptation for Spontaneous Speech Recognition. IEEE Proceedings of Acoustics, Speech, and Signal Processing (ICASSP '03), 2003, 1, I-236-I-239. https://doi.org/10.1109/ICASSP.2003.1198761

71. Zaman, S., Slany, W. Smartphone-Based Online and Offline Speech Recognition System for ROS-Based Robots. Information Technology and Control, 2014, 43(4), 371-380. https://doi.org/10.5755/j01.itc.43.4.5980

72. Zhai, C. Statistical Language Models for Information Retrieval. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2009.

Summary / Santrauka

Statistical language modeling involves techniques and procedures that assign probabilities to word sequences or, said in other words, estimate the regularity of the language. This paper presents basic characteristics of statistical language models, reviews their use in the large set of speech and language applications, explains their formal definition and shows different types of language models. A detailed overview of n-gram and class-based models (as well as their combinations) is given chronologically, by type and complexity of models, and in aspect of their use in different NLP applications for different natural languages. The proposed experimental procedure compares three different types of statistical language models: n-gram models based on words, categorical models based on automatically determined categories and categorical models based on POS tags. In the paper, we propose a language model for contemporary Croatian texts, a procedure how to determine the best n-gram and the optimal number of categories, which leads to significant decrease of language model perplexity, estimated from the Croatian News Agency articles (HINA) corpus. Using different language models estimated from the HINA corpus, we show experimentally that models based on categories contribute to a better description of the natural language than those based on words. These findings of the proposed experiment are applicable, except for Croatian, for similar highly inflectional languages with rich morphology and non-mandatory sentence word order.

Statistinis kalbos modeliavimas apima techniką ir procedūras, kurios įvertina žodžių sekų tikimybes arba, kitaip tariant, įvertina reguliarumą kalboje. Straipsnyje pristatomi pagrindiniai statistinių kalbos modelių bruožai, apžvelgiamas jų naudojimas didžiulėse kalbos taikymo aibėse, paaiškinamos jų formalios sąvokos bei pateikiami skirtingų kalbos modelių pavyzdžiai. Išsami n-gramomis ir klase pagrįstų modelių (bei jų kombinacijų) apžvalga pateikiama chronologiškai, pagal tipą ir modelių kompleksiškumą, taip pat pagal jų panaudojimą skirtinguose natūralios kalbos apdorojimo (NLP) taikymo skirtingoms natūralioms kalboms kontekstuose. Autorių siūlomas eksperimentinis būdas lygina tris skirtingus statistinius kalbos modelius: žodžiais grindžiamus n-gramos modelius, automatiškai nustatomomis kategorijomis grįstus kategorinius modelius bei kalbos dalies (POS) žymomis grįstus kategorinius modelius. Autoriai siūlo kalbos modelį moderniems tekstams, parašytiems kroatų kalba. Tai yra veiksmai, padedantys nustatyti geriausią n-gramą ir optimalų kategorijų skaičių. Kaip pavyzdį nagrinėjant Kroatijos Naujienų agentūros straipsnių (HINA) tekstyną, straipsnyje atskleidžiama, kaip siūloma procedūra reikšmingai sumažina kalbos modelio netikslumus. Naudodami skirtingus kalbos modelius, per eksperimentus su HINA tekstynu, autoriai parodo, kad kategorijomis grįsti modeliai padeda natūralią kalbą apibūdinti geriau, nei žodžiais grįsti modeliai. Be kroatų kalbos, pasiūlyto eksperimento rezultatai yra pritaikomi panašioms fleksinėms kalboms su turtinga morfologija ir ne griežtai nustatyta žodžių tvarka sakinyje.

ITC 4/46

Journal of Information Technology and Control
Vol. 46 / No. 4 / 2017
pp. 445-458
DOI 10.5755/j01.itc.46.4.18470
© Kaunas University of Technology

Service-Oriented Architecture Solution for ECG Signal Processing

Received 2017/05/21 | Accepted after revision 2017/10/23

http://dx.doi.org/10.5755/j01.itc.46.4.18470

Service-Oriented Architecture Solution for ECG Signal Processing

Tomas Blažauskas, Alvydas Muliuolis

Faculty of Informatics, Kaunas University of Technology, Studentu str 50, 51368 Kaunas, Lithuania
e-mails: tomas.blazauskas@ktu.lt, alvydas.muliuolis@ktu.lt

Liepa Bikulčienė, Eglė Butkevičiūtė

Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, Studentu str 50, 51368 Kaunas, Lithuania, e-mails: liepa.bikulciene@ktu.lt, egle.butkeviciute@ktu.lt

Corresponding author: tomas.blazauskas@ktu.lt

This paper introduces a distributed service-oriented system, which is developed to provide ECG (electrocardiogram) monitoring, analysis and storage services. The Service-Oriented Architecture system design is introduced for ECG signal transmission and processing. The implementation of cloud-based web-services and overall system architecture is described. The presented system includes a T-shirt with five electrodes intended for the acquisition of the signal. The ECG data for the experiment were recorded while the participant was moving. The signal replicates real conditions and the ECG data contain different high and low frequency noise. Therefore, this paper includes analysis of data filtering methods, model selection and ECG parameter calculation algorithms. The DWT algorithm was selected for the high frequency noise reduction and the BEADS method was used for trend removal. It was experimentally identified that these algorithms are effective and can be used in the system under development. The tests covering overall system were performed on an Amazon cloud computing infrastructure. The results are presented together with a discussion of various constraints of service-oriented performance.

KEYWORDS: Service-Oriented Architecture, high performance computing, ECG analysis, signal processing.

1. Introduction

With the increase of healthcare services in non-clinical environments using vital signs provided by wearable sensors, the need to process and analyse the physiological measurements are growing significantly [18]. An integrated solution of different sensors, smart interfaces, modelling, and data analysis technique should warrant that the created system is comfortable and effective for assessing the individuality and dynamics of a functional state during the daily-life activities and for the control of the exercise intensity.

To collect the data of the signal, a wearable device with five electrodes inserted into it, was used. Each of them contains different low and high frequency noise. This requires identifying and filtering out irrelevant sensor signals. Services built on top of the data collected by wearable devices need to provide useful information in real time calculations. That requires storage and processing of efficient (near) real-time data.

The extraction of parameters from the ECG signal is important, because the decision-making algorithm incorporates the evaluation of the ECG parameters. Recording the ECG signal at rest is an easy task. However, the records contain various origin noise during physical activity. Since storing the original signals takes too much space on a server [34], the parameter values may also be used as a compression of the ECG signal. This problem becomes important when the amount of the data increases. Future works will include methods to overcome the problem.

2. Related Works

Service-Oriented Architecture (SOA) systems for health monitoring are popular in health centres, clinics and smart home environments. These systems are used by elderly people, patients, sportsmen, etc. The most popular framework in a remote health monitoring system contains a three-tier architecture: a Wireless Body Area Network (WBAN), communication and networking system and a service layer. For instance, one of the suggestions was to use a system which contains wearable sensors to measure various physiological parameters, such as blood pressure and body temperature [6]. There are the health monitoring systems, utilizing similar cloud-based medical data storage. One of such systems [33] is used by doctors to access the stored data online using content service application. The electronic remote health monitoring systems sometimes can replace the conventional health care methods. However, integration of the Internet of Things (IoT) paradigm into these systems can further increase intelligence, flexibility and interoperability [31]. A device which utilizes the IoT is uniquely addressed and can be identified anytime and anywhere through the Internet. IoT-based devices in remote health monitoring systems can automatically connect and exchange the information with each other or with health institutes through the Internet. This allows to simplify the set-up and administration tasks significantly. In literature, there are some examples of the systems that can send automatic alarms to the nearest healthcare institute when supervised patient gets into critical accident [7]. In this paper, the proposed training system is also a type of health care system. In production, there is only one similar product, named *QardioCore* [29]. It evaluates the heart condition using ECG signal analysis and makes training intensity suggestions. Other products only evaluate the heart rate (HR). The heart rate (HR) is widely used to control workloads [15], but it does not reflect many other important physiological processes that ensure safety and effectiveness of physical activity. To personalize user health status, the monitoring system must efficiently process data of the sensors and visualize the holistic view (based on the complex systems theory [34]). Furthermore, the system should give feedback for the user about individualized intensity and duration of the training. The complexity of a signal (in a particular case, some characteristics of electrocardiogram (ECG) signal) may reflect human functional state and healthiness.

A variety of methods for the automated ECG analysis have been developed in the past few decades to simplify the monitoring task. The adaptive filtering [28], singular value decomposition (SVD) [12], independent component analysis (ICA) [37], neural networks [4], wavelet transform [14] are the most popular ones. However, almost all of these methods are not suitable for wearable sensors and real-time ECG filtering. Adaptive filtering and SVD methods are simple and fast in operations. Nevertheless, these methods fail if the data contain too much high and low frequency noise while the participant is moving. Wavelet transform and ICA methods are effective in eliminating the common ECG noises. However, both methods have high computational complexity, which is rather challenging to achieve in real-time ECG filtering. The methods with Neural Network algorithms require a lot of time for the ECG noise training and are not appropriate enough for individualized ECG filtering. In this paper, a Discrete Wavelet Transform (DWT) [8] method was selected for the high frequency noise reduction and the BEADS algorithm [26] for the trend removal.

3. Service-Oriented Architecture for ECG Monitoring

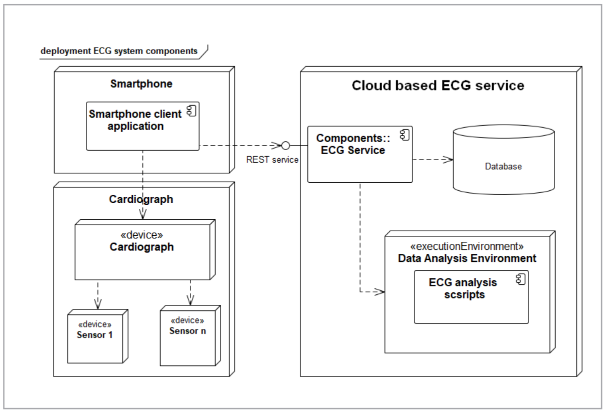
In this paper, the implementation of the minimal ECG monitoring system, which includes cloud-based ECG processing, was selected. It consists of the following functional units:

- A cardiograph device with its sensors, enabled to capture human state and provide information to a logging device located nearby;
- A logging device which collects information from the sensors and transmits it to a remote computer (server);
- A remote computer which receives, stores, analyses the data and provides feedback.

The implementation of these functional units and their relations are presented in Fig. 1. The arrows show the component usage relation.

Figure 1

Service-Oriented Architecture system components



The cardiograph node includes the cardiograph device and cardiograph sensors. The cardiograph acquires cardio signals using its sensors, converts them into digital data and transmits them to smartphone application using Bluetooth connection.

The logging device can be specially designed hardware and software or some general-purpose device which supports communication means required by cardiograph and remote computer. For this research, a smartphone has been chosen. Besides acting as a communication bridge between cardiograph and remote computer, the smartphone serves as the tool to provide feedback.

3.1. Services for ECG Monitoring System

The components of cloud-based services are described in this section, and it is the main part of the whole ECG system providing web services to store, analyse data and

provide the results, influencing all the other components. Cloud services consist of several software components, which can be used as distributed resources to provide the scalability of the system.

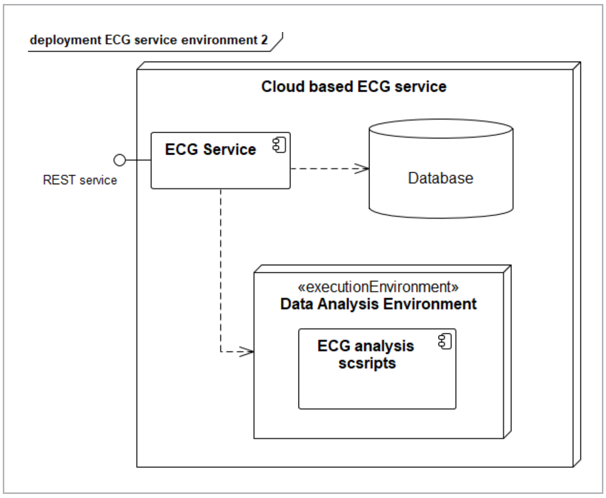
3.2. Server-Side Software Architecture

The simulation of complex cyber-physical systems is often regarded as expensive [38] in terms of the computational power. Therefore, the system which, on one side, is flexible and does not require exceptional computer power to handle service requests and, on the other side, may be implemented using distributed computer resources or other calculation capability (e.g. supercomputer for parallel calculations), was designed.

A system deployment scheme is provided in Fig. 2.

Figure 2

A web service system deployment scheme



The system uses two different software components to provide services for the following client software:

- *ECG Service*. This software is communicating with the client's software to provide actual services, based on the Representational State Transfer (REST) implementation. It is used to provide services, but does not perform data analysis operations itself. The main functionalities of the service are to receive ECG data, call Data Analysis

Environment algorithms and provide the results of the ECG data analysis to the client's software.

- *Data Analysis Environment*. The software performs the analysis of the ECG data provided.

The system can run many parallel data analysis environment instances and control them. Therefore, the system is scalable, i.e. the overall performance of the system can be improved by assigning more computation resources.

Scalability is very important for the system, because it gives the possibility to control the load and effectively use the cloud computing resources. For example, more computing power can be acquired for the short periods of time, when the system is handling an increased number of simulation requests. Particularly, using the cloud computing, dynamic scalability becomes more attractive and practical because of the unlimited resource pool [16].

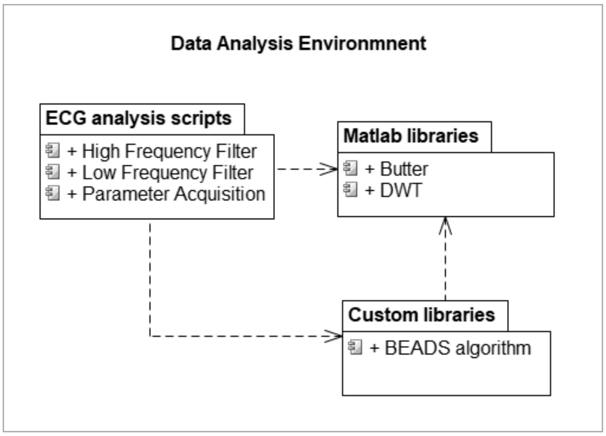
3.3. Data Analysis Environment

Data Analysis Environment (see Fig. 3) in the presented ECG monitoring system concept refers to a subsystem, consisting of MATLAB software and scripts, designed for the ECG parameter calculation.

All the scripts are included in the package named *Analysis Scripts.* This package uses MATLAB libraries as well as some custom libraries, helping to implement BEADS algorithm. The *Analysis scripts* package includes the following functionalities: *Low Frequency Filter, High Frequency Filter*, and *Parameter Acquisition.*

Figure 3

A general overview of the Data Analysis Environment



4. The ECG Processing Algorithms

The electrocardiogram (ECG) is a non-invasive measure of the cardiac electrical activity recorded by electrodes attached to the skin. Each heartbeat is an electrical impulse ("wave") traveling through the heart [3]. Usually, ECG signals are taken in stationary condition and the transmission of ECG often introduces noise due to poor channel conditions [3]. Despite signal variability, the signal noise comes from many sources, such as power line interfaces, muscular artefacts, electrode contact noise, baseline wanderings due to respiration, instrumentation noise generated by electronic devices, etc. [13], [19]. The ECG signal for medical and diagnostic purposes must be free from undesired disturbances and noise. Our task is to analyse these signals when the person is moving and doing exercise. The electrodes are embedded into T-shirts in order to reduce the noise by direct contact with the body surface. The ECG signal should be sampled with frequency of below 500 Hz, otherwise, it is not suitable for implementation in portable device (mobile) systems due to high computational demands [19].

There are three main problems in the initial ECG signal data analysis:

- Data trend removal. The trend in the ECG signal is understandable as low frequency noise (baseline). The trend appears as the result of muscle movement during an ECG recording.
- Noise reduction. In this case, it is understandable as a high frequency filter. High frequency noise is the result of electric uncertainty recording to ECG signal.
- Parameters calculation and data compression for the storage.

4.1. Data Filtering

The trend of the ECG signals is low when the person is not moving, and it could be removed by using low-order polynomial methods [2] or data could be subtracted from each time-series signal object. In this paper, the trend is considered as a low-frequency noise, which is not stationary or linear. It is assumed that noisy ECG data $y(n)$ can be modeled as:

$$y(n) = f(n) + x(n) + w(n), n = 0, \dots, N-1, \quad (1)$$

where f is a low-pass signal, x is a sparse-derivative signal, and w is a stationary white Gaussian noise [36]. For the data like $y(n)$, it is not suitable to use neither low-pass filtering, nor sparsity-based denoising methods. However, a combination of these two methods is found to be suitable [35].

The high-pass digital filter can be characterized by its transfer function which has the form:

$$H = H(z) = \frac{B(z)}{A(z)} = \frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_Nz^{-N}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_Nz^{-N}}, \qquad (2)$$

where the order of the filters is greater than N, [22]. The low pass filter L can now be defined as:

$$L = I - H, \qquad (3)$$

where I is the identity matrix. The combined methods of conventional low-pass filtering and sparsity-based denoising are usually complex and require long lasting calculations. The Baseline Estimation and Denoising with Sparsity (BEADS) [26] algorithm is used for trend removal of the ECG signals in the system. The BEADS trend removal algorithm is based on data baseline (or low-frequency noise wave) detection and its removal.

This algorithm requires some definitions. The filters L and H were taken with zero-phase, non-causal and recursive. There are two specification parameters of the filter: its order $2d$ and its cutoff frequency f_c [35], [26]. Moreover, by using the commutative property of Linear, Time-Invariant (LTI) systems, the filter H is defined as

$$H = BA^{-1}, \qquad (4)$$

where A and B are the finite matrices and they are not exactly commutative [35], [26]. Furthermore, if $x = v$, the cost function can be defined as:

$$G(x,v) = \frac{1}{2}\|H(y-x)\|_2^2 + x^T[\Gamma(x,r)] + \\ + \sum_{i=0}^{M}\left[\frac{\lambda_i}{2}(D_i x)^T[\Lambda(D_i v)](D_i x)\right] + c(v), \qquad (5)$$

where $c(v)$ is a scalar that does not depend on x and $\lambda_i \geq 0$ are regulation parameters, D_i is the order-i difference operator, $\Gamma(x, r)$ is a diagonal matrix that depends on asymmetric parameter r [8]. Furthermore,

$$[\Lambda(D_i v)]_{n,n} = \frac{\varphi'(v_n)}{v_n} \qquad (6)$$

depends on penalty function φ. For example, if $\varphi(x) = |x|$, then $[\Lambda(D_i v)] = \frac{1}{D_i v} sign(D_i v)$ [35], [26].

The cost function $G(x, v)$ is needed in order to find the data baseline. Minimizing $G(x, v)$ with respect to the following solution:

$$x = \left(H^T H + \sum_{i=0}^{M} \lambda_i D_i^T [\Lambda(D_i v)] D_i\right)^{-1} H^T H y = \\ = AQ^{-1}B^T BA^{-1}y, \qquad (7)$$

where $Q = B^T B + A^T(\sum_{i=0}^{M} \lambda_i D_i^T [\Lambda(D_i v)] D_i)A.$

The T-shirts have five electrodes that are inserted into different places and three ECG leads are recorded from these electrodes. All recorded signals from the ECG registration device are sent as a data set to the smart phone. There are many different exercises that participant might perform, so the trend might also vary.

ECG simulator data were used for calculation verification to check if BEADS algorithm works fine. Each value of the signal is taken every 2 ms at 500 Hz frequency. The simulator generates 150 bpm sinusoids (pure ECG signal). The low and high frequency noise was added to the simulated electrocardiogram. The BEADS algorithm was compared with other algorithms (Butterworth filter [17], Fast Fourier Transform algorithm (FFT) [10] and Finite Impulse Response filter (FIR) [21]). In addition, random Gaussian values with scalar 10 (which means the signal-to-noise ratio per sample, dB) were enclosed as a high frequency noise and the sinusoid:

$$f(t) = 0.3 \cdot \sin(0.9 \cdot \pi \cdot t) + 0.001 \cdot rand$$

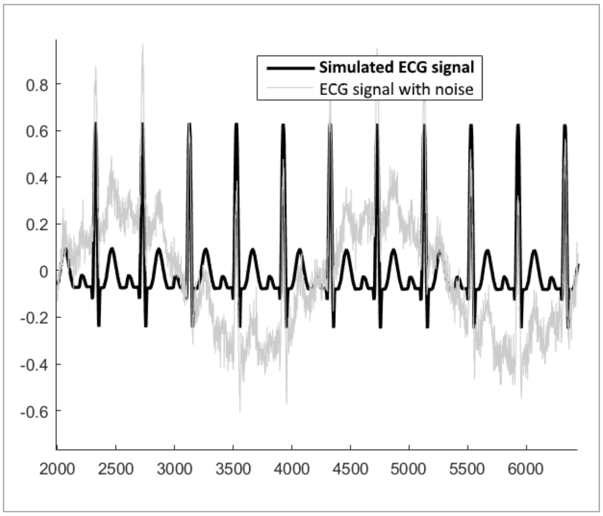
as a low frequency noise (trend). Here *rand* is random scalar values with standard normal distribution. The simulated ECG signal is presented in Fig. 4.

The BEADS algorithm has the asymmetry parameter which could vary. During analysis, it was noticed that the best value is from 1 to 6 due to low Root Mean Square Error (RMSE) values.

Furthermore, the 9-th order Butterworth filter [17] was selected for error comparison. The normalized frequency may vary from 0 to 1. However, the values from 0 to 0.5 for low-pass filtering are taken. The RMSE was taken as a measurement to find the best frequency in the BEADS and Butterworth methods.

Figure 4

Simulated ECG signal



The results are presented in Table 1. The best frequency from the table is $f_c = 0.01$. The Butterworth filter causes small RMSE values. However, the BEADS algorithm is more accurate (Table 1).

Table 1

A comparison of BEADS and Butterworth

fc	RMSEBEADS	RMSEButterworth
0.01	0.1	0.15
0.02	0.12	0.13
0.04	0.14	0.14
0.05	0.14	0.15
0.07	0.15	0.16
0.08	0.15	0.16
0.1	0.15	0.16
0.11	0.15	0.17
0.13	0.16	0.17
...	...	...
0.5	0.16	0.17

Figure 5

The trend removal from the ECG signal using the BEADS algorithm: a) original signal; b) baseline (black) and original signal (grey); c) signal after the BEADS algorithm; d) residuals

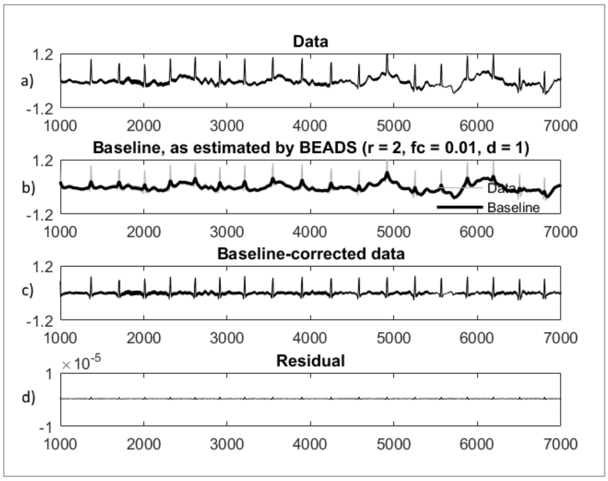
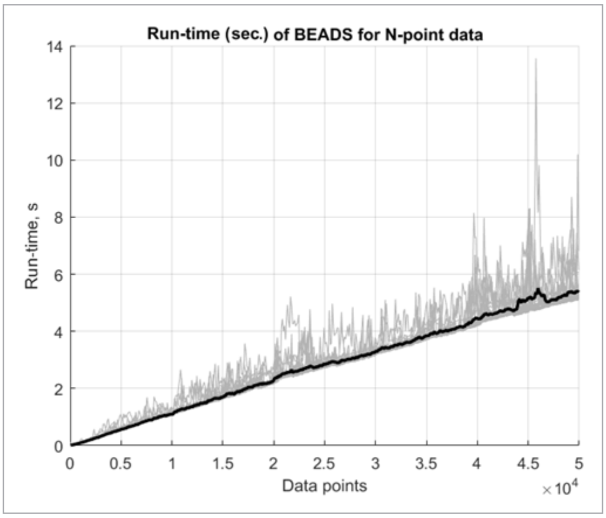


Fig. 5 shows how the BEADS algorithm works when a person with a special T-shirt is walking. The asymmetry ratio $r = 2$, filter order $d = 1$, and filter cut-off frequency $f_c = 0.01$ are the values in this example.

The BEADS algorithm works fine and the residuals (original data minus baseline minus baseline-corrected data) are low (see Fig. 5). However, the complexity is also important because the real-time results are necessary during the exercises.

Figure 6

Complexity of BEADS algorithm



The run-time of BEADS for N-point data is presented in Fig. 6. Sometimes the run time may reach as much as 30 s. The reason behind this variability is that BEADS calculates coefficients of the high-pass filter, while cost function (1.5) does not reach the desired low value. The inverse matrices (see (1.7)) are found during these calculations. In some cases, it is complex to calculate the inverse matrices and it requires more time. This causes the peaks, which can be seen in Fig. 6. The run-time average of 50 iterations (thicker line) shows that the complexity of the BEADS algorithm is linear.

The BEADS algorithm was used to reduce the low frequency noise (trend removal). The ECG signal also has a high frequency noise (as it was mentioned before) which was not reduced by the BEADS algorithm.

Fast Fourier Transform (FFT) is one of the most popular methods to reduce high frequency noise. However, this technique fails to provide the information about exact location of frequency components in time [10]. The Discrete Wavelet Transform (DWT) algorithm was chosen for ECG signal analysis. The DWT algorithm is appropriate to be used for non-stationary signals (like ECG) because it decomposes the signal into a frequency time scale [13]. This method is based on wavelet transformation, in which copies of the main pattern ("mother wavelet") are scaled and shifted [13]. Each analysing wavelet has its own time duration and time location [10]. The mother wavelet in DWT is discretized and can be written as follows:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{8}$$

where b is a translation parameter of prototype wavelet $\psi(t)$, and a is a scale parameter, obtained by means of dilation [14], [13].

The optimal wavelet function must be selected in order to achieve the best noise free signal. Furthermore, the best threshold selection rule and decomposition level should be found. The DWT algorithm has the following three steps [8], [5]:

- The DWT is applied on a noisy signal.
- The thresholding process is applied. The wavelet coefficients are filtered by throwing them away (resetting to zero) during this process. The idea of this thresholding process is based on an assumption that the noise is generated by small-value wavelet coefficients, while the large coefficients compose the actual signal.
- The remaining coefficients are back-converted in time domain (Inverse Discrete Wavelet Transform (IDWT)).

The DWT algorithm contains many arithmetic operations and requires a large memory storage. This is not desirable for real-time calculations. It is not practical to directly compute the DWT for the entire signal. That is why the three-level wavelet de-noising process [8] was used.

The same simulated data (see Fig. 4) were taken for the comparison of high-pass filters and parameter analysis. However, in this case, the data have no trend. It was removed in the previous step with the BEADS algorithm.

The right wavelet should be selected to find the best solution with the DWT algorithm. The Daubechies wavelets, based on the work of Ingrid Daubechies, are a family of orthogonal wavelets defining a discrete wavelet transform, [13]. Each wavelet type of this class has a scaling function generating an orthogonal multiresolution analysis. Each Daubechies wavelet generates different filtering coefficients.

The higher order leads to better errors but it requires long lasting calculations. If more calculations are required, the model might become insufficient for real time analysis. In this paper, 3400 data points were analysed to find the best wavelet. The RMSE was used to measure the accuracy of DWT with different wavelets (Table 2). The timing is important for real-time data analysis. For this purpose, the calculation time is presented in Table 2.

The DWT algorithm gives the optimal results with wavelets "db11" and "db12": the RMSE values are similar with errors of higher order Daubechies coefficients. Further calculations reduce RMSE slightly. In addition, Daubechies wavelet with the higher order causes minor changes in duration time. However, as was mentioned before, the higher order of these coefficients makes the model more complex and that is undesirable for the real-time calculations. The "db11" will be used for the calculations below.

Table 2

The errors of the DWT algorithm and calculation time for different Daubechies wavelets

Daubechies	RMSE	Duration, s
'db5'	0.0579	0.4900
'db6'	0.0574	0.6110
'db7'	0.0570	0.8690
'db8'	0.0568	0.9680
'db9'	0.0564	0.7060
'db10'	0.0562	0.7760
'db11'	0.0559	0.8279
'db12'	0.0561	0.6070
'db13'	0.0558	1.0760
'db14'	0.0561	1.7529
'db15'	0.0558	0.7558
'db16'	0.0557	1.1080
'db17'	0.0555	0.5440
'db18'	0.0555	0.7219
'db19'	0.0555	0.8588
'db20'	0.0555	0.6396
'db21'	0.0555	0.6336
'db22'	0.0555	0.7249

The DWT algorithm was compared with Butterworth to see which algorithm is more efficient. The DWT algorithm and Butterworth filter give different results. To compare which one was better, the RMSE values were calculated:

$RMSE_{Butterworth} = 0.1206,$

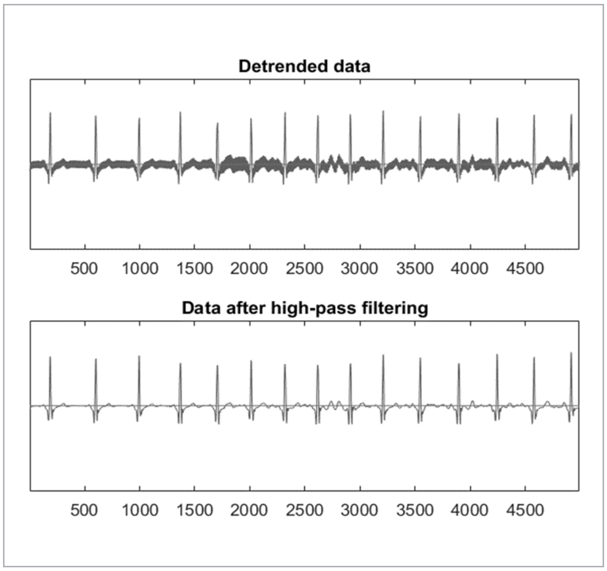
$RMSE_{DWT} = 0.0559.$

RMSE shows that the DWT algorithm reduces the noise better (with lower errors).

A DWT filtering example with real data (from Cardioscout multi ECG registering device) is shown in Fig. 7. The first graph shows the ECG data when the participant is walking (the trend is already removed), while the second part of this figure shows denoised data with DWT.

Figure 7

Detrended ECG signal and filtered with DWT



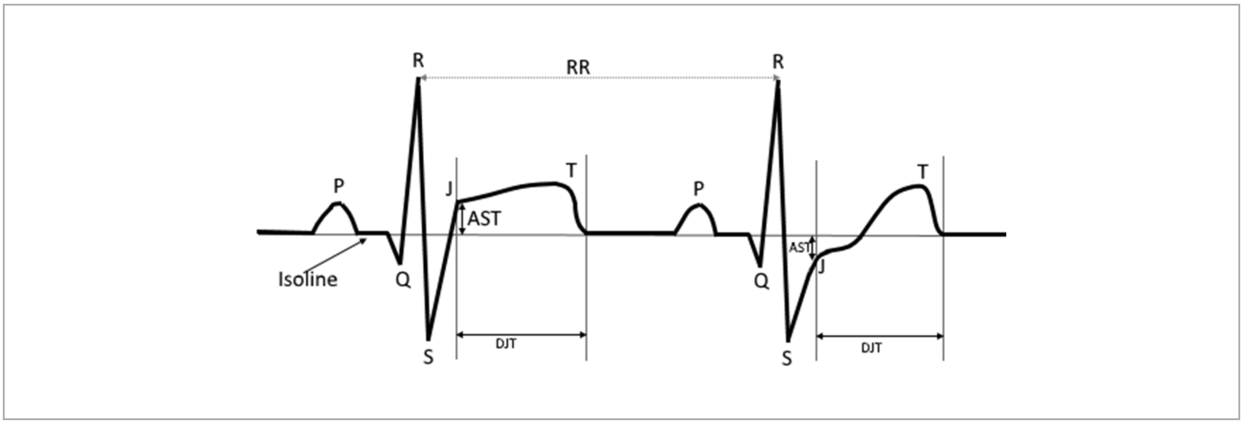
4.2. ECG Parameters Calculation

Automatic R-wave recognition is the first processing task of the ECG signal. Later, the QRS complex and other ECG parameters can be measured. The QRS complex (see Fig. 8) includes recognition of their position in time and time interval between them [23]. The QRS complex has the best ratio of signal/noise characteristics. The main detection algorithms of the QRS complex are based on the first or both first and second order derivative calculation for signal and digital filtering. Most of the time local extreme signal detection logic has additional rules that can be applied to reduce the number of false crawls. Wavelet transform [20], filter banks [1], neural networks, adaptive filters, Hidden Markov Models (HMM), mathematical morphology operators, genetic algorithms, Hilbert, length and energy transformations, syntactic methods, MODB algorithms (based on several signal values of derivative product) [23] can also be used for QRS detection. Most of these methods require substantial computing resources, have a big delay, are sensitive to noise and sudden signal change [9], so the commonly used algorithm remains the Pan-Tomkinson algorithm [27] based on the calculation of derivatives.

As was mentioned before, the ECG is used to measure the heart rate and regularity of heartbeats. It shows

Figure 8

ECG parameters



the size and position of chambers. Heart diseases or damages could also be detected [3] from ECG. The ECG signal has a distinct and characteristic shape which is given in Fig. 8.

The purpose of this part of the ECG processing is to find P and T waves, QRS complex, AST and DJT parameters (see Fig. 8). Their calculation is based on local extrema and twist points search in RR interval. These parameters will be used in future works to design the decision-making algorithm. The following parameters were chosen by recommendations of sport's medicine specialists:

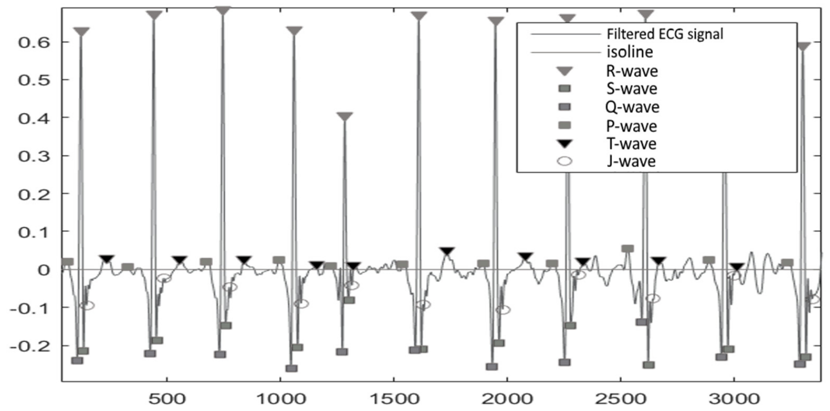
- The P wave appears because of the contraction in the left and right atria. This wave may be even absent from some ECG recordings [24]. The normal shape of a P wave does not include any peaks. Furthermore, it can be positive, negative or biphasic [25].
- The QRS complex duration is necessary for the many medical instruments. This complex is made by the contraction of the left and right ventricles. Q, R, and S waves involve more muscle mass and are stronger than the P waves. That is why they have larger fluctuation in the graph [10], [30].
- The T wave is necessary for a variety of diagnostic tasks (acute coronary syndrome, acute myocardial infarction, and potentially fatal arrhythmias). This wave is caused by the repolarization of the ventricles [10], [11].
- The JT duration (DJT) reflects the metabolism of the heart. However, when the QRS duration is increased (which contributes to QT prolongation), it has been proposed that the JT interval duration is a more appropriate measure of ventricular repolarization than the QT duration.
- The ST segment represents the interval between ventricular depolarization and repolarization. This segment in the first stage is the flat, isoelectric section of the ECG between the end of S wave (the J point) and the beginning of the T wave. Myocardial ischaemia, or infarction, is the most important cause of the ST segment abnormality (elevation or depression). This parameter is noted as AST in Fig. 8 [10].

All calculations during the parameter search in Fig. 9 were executed with detrended and high-pass filtered data. In some cases, it is difficult to find the parameters because small parts of the data may still contain noise. These values are taken as average from three previous parameter values.

The effective ECG signal compression demand is quite high, assuming the fact that every year millions of electrocardiograms are recorded and remote transmission that uses the network grows. Effective data compression is needed in many practical applications: ECG data storage, ambulatory monitoring systems, ECG data transmission network. The data compression techniques are divided into those containing compressed data with reconstructed into the original signal and the methods with higher compression ratio achieved by allowing certain errors in reconstructing the signal [32]. The efficiency of the ECG signal

Figure 9

ECG waves (QRS complex, P and T waves, J wave)



compression can be measured by the ratio between the compressed data and primary data (compression ratio CR), primary and compressed data size, time to compress and restore data and error, often referred to as a percentage of the average square difference (percent mean-square difference – PRD).

5. Performance of the Cloud-Based Solution

The calculation of the ECG parameters is a resource intensive task. It is necessary to evaluate the performance of the solution to see if it can be applied practically. Two experiments were performed to see how the system handles the load.

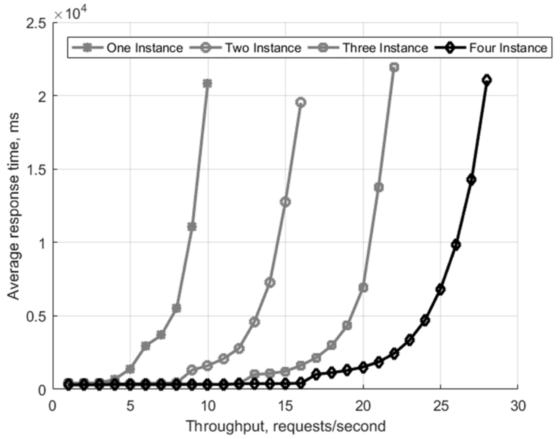
The Amazon cloud computing services were used with preconfigured four machines in the experiments. Each machine contained one virtual CPU (Intel Xeon CPU E5-2676 v3, single core, 2.40 GHz), and 1GB memory. The load balancing mechanism is provided by the Amazon services. The Apache JMeter tool was used for the experiments.

The JMeter tool was set up to create thread group with ramp-up period equal to 10 s and duration set to 180 s.

The number of threads (users) was increased from 10 to 400 with step 10. Each request was sending 42.5 kilobytes. Requests were delayed with constant timer, which was set to 10000 ms. The average response time was calculated from interval [60s; 120s]. This interval was chosen, because the system must have become stable before calculating average response time. A straight forward approach was used in the first experiment – the requests were generated continuously, increasing the amount of the requests per second, until the system starts degrading. The results are pro-

Figure 10

Performance of the solution for continuously generated requests



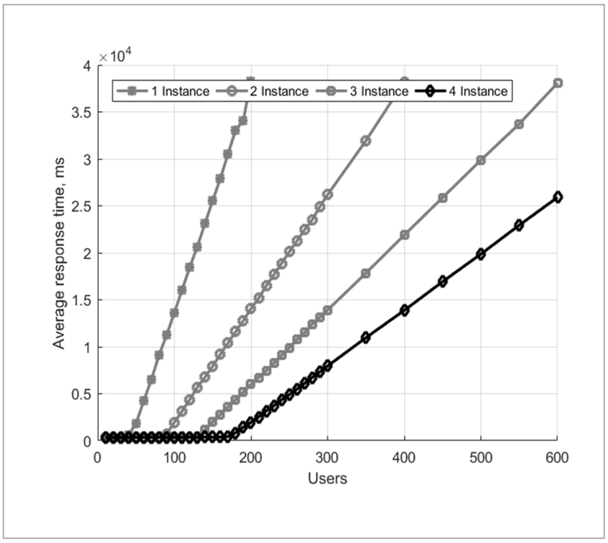
vided in Fig. 10. It shows that the solution, distributed on 4 machines (with given hardware specifications) can handle up to 16 requests per second. The system degrades with bigger load and becomes unusable. To determine how many users are needed to produce the desired requests per second (RPS), the following formula can be used:

$$Users = RPS \cdot \frac{Average\ Response\ Time}{1000} \quad (9)$$

This metric allows to see how the system performs on stress loads; it does not show how many users our system can handle when implementing our use case. In our use case, each user sends a request every 10 seconds. Therefore, this behaviour was simulated using the Apache JMeter tool. The results are provided in Fig. 11. It shows that the solution, distributed on 4 machines (4 instances) can handle up to 170 users without degrading behaviour. The system degrades linearly with bigger number of users.

Figure 11

Performance of the solution when simulating practical application



6. Conclusions

Medical systems are increasingly being implemented as cloud-based services to use the benefits of cloud computing. Our task was to study the effectiveness of cloud-based solution designed for near real-time ECG signal analysis to replace legacy systems. For that purpose, a system, which includes service-oriented architecture solution as well as algorithms for the ECG signal analysis, was created.

The study of the implemented algorithms shows that:

- The BEADS algorithm with normalized frequency f_c = 0.01 and asymmetry r from 1 to 6 is the best trend removal method for analysed ECG signal.
- The DWT algorithm was selected for ECG signal noise reduction. The best results were found with Daubechies wavelet 'db11' and they caused just 0.0559 RMSE error.

The study of the cloud-based solution shows that:

- The performance of the proposed cloud-based solution with four dedicated virtual machines allows serving approximately 250 concurrent users for the given use case.
- The efficiency is sufficient for the most use cases, although the improvement on architectural and algorithm levels is needed.

The future works will contain data compression, automated health evaluation and training methodology system using decision making algorithms.

Acknowledgment

This work was supported by a grant from the Agency for Science, Innovation and Technology (MITA) regarding Eureka project ITEA2 CareWare 13034.

References

1. Afonso, V. X., Tompkins, W. J., Nguyen, T.Q., Luo, S. ECG Beat Detection Using Filter Banks. IEEE Transactions on Biomedical Engineering, 1999, 46(2), 192-202. https://doi.org/10.1109/10.740882
2. Akshay, N., Jonnabhotla, N. A. V., Sadam, N., Yeddanapudi, N. D. ECG Noise Removal and QRS Complex Detection Using UWT. ICEIE, 2010, 2, 438-442. https://doi.org/10.1109/ICEIE.2010.5559812

3. AlMahamdy, M. Riley, D. Performance Study of Different Denoising Methods for ECG Signals. The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2014), Elsevier, Procedia Computer Science 37, 2014, 325-332. https://doi.org/10.1016/j.procs.2014.08.048
4. Anuradha, B., Veera Reddy, V. C. ANN for Classification of Cardiac Arrhythmias. ARPN Journal of Engineering and Applied Sciences, 2008, 3(3), 1-6.
5. Arvinti, B., Costache, M., Toader, D., Oltean, M., Isar, A. ECG Statistical Denoising in the Wavelet Domain. IEEE, 9th International Symposium on Electronics and Telecommunications (ISETC), 2010, 307-310. https://doi.org/10.1109/ISETC.2010.5679269
6. Babu, S., Chandini, M., Lavanya, P., Ganapathy, K., Vaidehi, V. Cloud-Enabled Remote Health Monitoring System. International Conference on Recent Trends in Information Technology (ICRTIT), 2013, 702-707. https://doi.org/10.1109/ICRTIT.2013.6844286
7. Bui, N., Zorzi, M. Health Care Applications: A Solution Based on the Internet of Things. Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technology, (ISABEL'11), New York, NY, USA: ACM, 2011, 131:1-131:5. https://doi.org/10.1145/2093698.2093829
8. Chen, S.W., Chen, Y. H. Hardware Design and Implementation of Wavelet De-Noising Procedure for Medical Signal Preprocessing. Sensors, 2015, 15, 26396-26414. https://doi.org/10.3390/s151026396
9. Clifford, G. D., Azuaje, F., McSharry, P. Advanced Methods and Tools for ECG Data Analysis. Artech House Publishers, Norwood, 2006.
10. Elbuni, A., Kanoun, S., Elbuni, M., Ali, N. ECG Parameter Extraction Algorithm Using (DWTAE) Algorithm. International Conference on Computer Technology and Development, 2009, 57-62.
11. Elgendi, M., Eskofier, B., Abbott, D. Fast T Wave Detection Calibrated by Clinical Knowledge with Annotation of P and T Waves. Sensors, 2015, 15(7), 17693-17714. https://doi.org/10.3390/s150717693
12. Gao, D. Madden, M. Bayesian ANN Classifier for ECG Arrhythmias Diagnostic System. Proceedings of IEEE International Joint Conference on Neural Network, 2003, 2454-2459.
13. Garg, G., Singh, V., Gupta, J. R. P., Mittal, A. P. Optimal Algorithm for ECG Denoising Using Discrete Wavelet Transforms. 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2010. https://doi.org/10.1109/ICCIC.2010.5705839
14. German-Sallo, Z., Lutean, C. Wavelet Packet Transform Based ECG Signal Filtering Implemented in Reconfigurable Hardware Structure. Scientific Bulletin of the «Petru Maior» Universtity of Tirgu Mures, 2014, 11(1), 42-45.
15. Hoffmann, K., Wiemeyer, J., Hardy, S., Göbel, S. Personalized Adaptive Control of Training Load in Exergames from a Sport-Scientific Perspective. In: Göbel S., Wiemeyer J. (Eds.) Games for Training, Education, Health and Sports. GameDays 2014, Lecture Notes in Computer Science, 8395, 129-140. https://doi.org/10.1007/978-3-319-05972-3_14
16. Hölzl, M., Gabor, T. Continuous Collaboration: A Case Study on the Development of an Adaptive Cyber-Physical System. IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS), 2015, 19-25.
17. Jagtap, S. K., Uplane, M. D. The Impact of Digital Filtering to ECG Analysis: Butterworth Filter Application. International Conference on Communication, Information & Computing Technology (ICCICT), 19-20 Oct., 2012.
18. James, D. A., Petrone, N. Sensors and Wearable Technologies in Sport: Technologies, Trends. and Approaches for Implementation SpringerBriefs in Applied Sciences and Technology, 2016. https://doi.org/10.1007/978-981-10-0992-1
19. Jokic, S., Krco, S., Delic, V., Sakac, D., Jokic, I., Lukic, Z. An Efficient ECG Modeling for Heartbeat Classification. 10th Symposium on Neural Network Applications in Electrical Engineering, 2010, 73-76. https://doi.org/10.1109/NEUREL.2010.5644105
20. Kadambe, S., Murray, R., Boudreaux-Bartels, G. F. Wavelet Transform-Based QRS Complex Detector. IEEE Transactions on Biomedical Engineering, 1999, 46(7), 838-848. https://doi.org/10.1109/10.771194
21. Karam, M., Khazaal, H. F., Aglan, H., Cole, C. Noise Removal in Speech Processing Using Spectral Subtraction. Journal of Signal and Information Pro-

cessing, 2014, 5, 32-41. https://doi.org/10.4236/jsip.2014.52006

22. Kaur, M., Singh, B., Ubhi, J. S., Rani, S. Digital Filteration of ECG Signals for Removal of Baseline Drift. International Conference on Telecommunication Technology and Applications, Singapore, 2011, 5, 105-109.

23. Köhler, B.-U., Hennig, C., Orglmeister, R. The Principles of Software QRS Detection. IEEE Engineering in Medicine and Biology, 2002, 42-57.

24. Mehta, S. S., Lingayat, N. S. Detection of P and T Waves in Electrocardiogram. Proceedings of the World Congress on Engineering and Computer Science, 22-24 Oct., 2008.

25. Messaoud, M. B., Khelil, B., Kachouri, A. Analysis and Parameter Extraction of P Wave Using Correlation Method. The International Arab Journal of Information Technology, 2009, 6(1), 40-46.

26. Ning, X., Selesnick, I. W., Duval, L. Chromatogram Baseline Estimation and Denoising Using Sparsity (BEADS). Chemometrics and Intelligent Laboratory Systems. Polytechnic School of Engineering, 2014, 139, 156-167.

27. Pan, J., Tompkins, W. J. A Real-Time QRS Detection Algorithm. IEEE Transactions on Biomedical Engineering, BME-32, 1985, 3, 230-236.

28. Principe, J. C., Euliano V., Lefebvre W. C. Neural and Adaptive Systems: Fundamentals Through Simulations. John Wiley & Sons, Inc., 2000.

29. Qardiocore Wearable ECG Unlike Any Other. https://www.getqardio.com/qardiocore-wearable-ecg-ekg-monitor-iphone/

30. Qiu, P., Ray Liu, K. J. A Robust Method for QRS Detection Based on Modified P-Spectrum. IEEE ICASSP, 2008, 501-504. https://doi.org/10.1109/ICASSP.2008.4517656

31. Ray, P. Home Health Hub Internet of Things (H3iot): An Architectural Framework for Monitoring Health of Elderly People. International Conference on Science Engineering and Management Research (ICSEMR), 2014, 1-3. https://doi.org/10.1109/ICSEMR.2014.7043542

32. Ranjeet, K., Kumar, A., Pandey, R. K. ECG Signal Compression Using Different Techniques. Advances in Computing, Communication and Control, CCIS 125, 2011, 231-241.

33. Rolim, C., Koch, F., Westphall, C., Werner, J., Fracalossi, A., Salvador, G. A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions. Second International Conference on eHealth, Telemedicine, and Social Medicine, ETELEMED'10, 2010, 95-99. https://doi.org/10.1109/eTELEMED.2010.19

34. Roshan, J. M., Rajendra, A., Hojjat, A. Current Methods in Electrocardiogram Characterization. Computers in Biology and Medicine, 2014, 48, 133-149. https://doi.org/10.1016/j.compbiomed.2014.02.012

35. Selesnick, I. W., Graber, H. L., Pfeil, D. S., Barbour, R. L. Simultaneous Low-Pass Filtering and Total Variation Denoising. IEEE Transactions on Signal Processing, 2014, 62(5), 1109-1124. https://doi.org/10.1109/TSP.2014.2298836

36. Tarvainen, M., Ranta-aho, P.O., Karjalainen, P. A. An advanced detrending method with application to HRV analysis. IEEE Tram. Biomed. Eng. vol. 49, pp. 172-175, 2002. https://doi.org/10.1109/10.979357

37. Thakor, N. V., Zhu, Y.-S. Applications of Adaptive Filtering to ECG Analysis: Noise Cancellation and Arrhythmia Detection. IEEE Transactions on Biomedical Engineering, 1991, 38(8), 785-794. https://doi.org/10.1109/10.83591

38. Xiao, T., Fan, W. Modeling and Simulation Framework for Cyber Physical Systems. In: Kim, J.-H., Lee, K., Tanaka, S., Park, S.-H. (Eds.), Advanced Methods, Techniques, and Applications in Modeling and Simulation. Proceedings in Information and Communications Technology, 4, Springer, Tokyo, 2012, 105-115. https://doi.org/10.1007/978-4-431-54216-2_13

Summary / Santrauka

This paper introduces a distributed service-oriented system, which is developed to provide ECG (electrocardiogram) monitoring, analysis and storage services. The Service-Oriented Architecture system design is introduced for ECG signal transmission and processing. The implementation of cloud-based web-services and overall system architecture is described. The presented system includes a T-shirt with five electrodes intended for the acquisition of the signal. The ECG data for the experiment were recorded while the participant was moving. The signal replicates real conditions and the ECG data contain different high and low frequency noise. Therefore, this paper includes analysis of data filtering methods, model selection and ECG parameter calculation algorithms. The DWT algorithm was selected for the high frequency noise reduction and the BEADS method was used for trend removal. It was experimentally identified that these algorithms are effective and can be used in the system under development. The tests covering overall system were performed on an Amazon cloud computing infrastructure. The results are presented together with a discussion of various constraints of service-oriented performance.

Straipsnyje pristatoma saityno paslaugų sistema, skirta EKG (elektrokardiogramos) signalo stebėsenai, analizei ir saugojimui. Pateikiama ir aprašoma į saityno paslaugas orientuota sistemos architektūra, realizuojanti EKG signalo perdavimą ir apdorojimą debesų kompiuterijos priemonėmis. Signalo registravimui naudojami išmanūs marškinėliai su penkiais integruotais elektrodais. EKG duomenys buvo įrašomi treniruotės metu, kai eksperimento dalyvis atlikinėjo įvairius fizinius pratimus. Tokiu būdu gautas signalas yra panašus į realiomis sąlygomis išgaunamą signalą, kuris turi aukšto ir žemo dažnio triukšmą. Straipsnyje aprašomi signalo filtravimo algoritmai, atliekama jų analizė ir aprašomi EKG parametrų skaičiavimo algoritmai. Aukšto dažnio triukšmo sumažinimui parinktas DWT algoritmas, o žemo dažnio triukšmo pašalinimui – BEADS metodas. Eksperimento metu buvo nustatyta, kad šie algoritmai yra efektyvūs ir tinkami kuriamai sistemai. Buvo atlikti našumo testai sistemai, įdiegtai „Amazon cloud“ debesų kompiuterijos infrastruktūroje. Straipsnyje pristatomi rezultatai ir diskusija, susijusi su tokios realizacijos sistemos našumu.