TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja
Text Analysis and Knowledge Engineering Lab
Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

BACHELOR THESIS num. 5329

Sentiment Summarization from Student Course Questionnaires

Filip Šaina

Zagreb, June 2017

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING
BACHELOR THESIS COMMITTEE

Zagreb, 3 March 2017

BACHELOR THESIS ASSIGNMENT No. 5329

Student:	**Filip Šaina (0036479300)**
Study:	Computing
Module:	Computer Science

Title: **Sentiment Summarization from Student Course Questionnaires**

Description:

Sentiment summarization is a natural language processing task that combines document summarization and sentiment analysis. The main idea is to generate summaries of subjective texts that account for sentiment toward the different aspects of a product or service. Sentiment summarization systems enable the analysis of large amount of subjective texts, e.g., user reviews or comments on social networks. One interesting and useful application of sentiment summarization is the analysis of student course questionnaires.

The topic of this thesis is sentiment summarization from student course questionnaires in Croatian language. Do a literature survey on extractive document summarization, sentiment polarity classification, and sentiment summarization based on machine learning. Devise and implement a method for sentiment summarization from student course questionnaires that can identify the various aspects of a course, group comments pertaining to similar aspects, and classify their sentiment polarity. Compile a suitable dataset for training and evaluating the model, derived from real student questionnaires. Devise a suitable and intuitive way for visualizing the sentiment summarization results. Carry out an experimental evaluation of the method on test data. All references must be cited, and all source code, documentation, executables, and datasets must be provided with the thesis.

Issue date:	10 March 2017
Submission date:	9 June 2017

Mentor:

Associate Professor Jan Šnajder, PhD

Committee Chair:

Full Professor Siniša Srbljić, PhD

Committee Secretary:

Assistant Professor Tomislav Hrkać, PhD

I would like to thank my loving mother Ida and father Ivan for their continuous support throughout the duration of my undergraduate education.

I would also like to thank my mentor – prof. dr. sc. Jan Šnajder, for his guidance and feedback throughout the duration of this bachelors thesis.

CONTENTS

1. Introduction

Through the years, the amount of university student questionnaires[1] has met the point of becoming a valuable dataset to explore. Subjective, sentiment filled feedback from students provides a valuable feedback to professors and in the same time proves to be an interesting dataset to explore. Currently, due to large number of student questionnaires being made every semester quarter, a potentially helpful high-level overview of the overall sentiment is missed, or obtained with time exhaustive reading by professors.

Natural language processing (NLP) is a scientific discipline resulting from three major overlapping fields of study: computer science, linguistics, and artificial intelligence. Recent breakthroughs in machine learning[2] provide new tools for modeling and reasoning about natural human language language.

State-of-the-art overview is available in the survey by Schouten i Frasincar (2016) in which we observe three general branches of aspect-level sentiment analysis: Aspect detection, Sentiment analysis, Joint aspect detection and sentiment analysis. Where the filed of study taxonomy is derived into multiple approaches, as visible in 5.1. This thesis focuses on supervised and unsupervised Machine learning approaches on all three fields.

In this bachelor thesis a model is devised, implemented, and tested to apply aspect based sentiment analysis (ABSA) over student questionnaires in order to extract main aspects and provide a sentiment label over every aspect. The language in which this dataset is written is Croatian. To the best of the authors knowledge, this is the first such system implemented both for English and Croatian. Due to the unstructured nature and small size of the dataset an unsupervised method for aspect exaction proves to perform best, while for sentiment classification a standard supervised SVM classification is applied. To simplify system usage a simple web application is implemented.

[1] `http://www.fer.unizg.hr/`

[2] Machine learning is a subfield of computer science.

Thesis Outline

The reminder of this thesis is organized as follows:

- Chapter 2 talks about the current state of Aspect-based sentiment analysis;
- Chapter 3 provides an overview of the dataset and the labeled dataset split as well how the word embeddings were constructed;
- Chapter 4 talks about the following three subtasks: Aspect term labeling, Aspect sentiment classification, and Aspect-sentiment aggregation. All implemented subtask models are described along with their respective evaluations scores on the given datasets;
- Chapter 5 talks about the web application developed for using the system;
- Chapter 6 concludes the thesis and outlines ideas for future work.

2. Aspect-based Sentiment Analysis

2.1. Introduction

Throughout this thesis, the main focus will be on aspect and sentiment models. To provide a common ground, a definition of what aspects and sentiments are is provided here.

Aspect

As discussed by Liu (2010), an aspect or feature is an attribute of an entity, for example the slide quality is an aspect of a lecture, screen of a laptop etc. The advantage of analyzing text over aspects is the possibility of determining subtle differences in a expression regarding different entities. As entities can be described trough multiple aspects, an aspect can therefore be assigned multiple sentiments for every aspect-feature.

Sentiment

Sentiment analysis is a field of study that primarily quantifies subjective information in various texts. In the field of natural language processing, the primarily study of sentiment is made trough classification tasks of given text into subjective and/or objective classes Pang et al. (2008). Another major filed of interest is polarity classification where a given text is classified in terms of its sentiment polarity (e.g., positive, neutral, negative). This thesis will focus on sentiment polarity classification, as discussed later in Chapter 4, both through subjectivity-objectivity classification and direct classification.

2.2. Aspect-based Sentiment Analysis

The amount of vast daily community-generated data in form of reviews, topic comments (e.g., electronics, books, etc.) that touch on multiple aspects of an entity (e.g., screen size, weight, lecture quality, etc.) provide an incredible opportunity for creting an aggregated high-level community stance towards those entities.

Aspect-based sentiment analysis (ABSA) systems, for a given text (e.g., lecture review), extract the most prominent aspects (features) that model the overall topic (e.g., university course). In this thesis, following the task decomposition proposed in the work of Pavlopoulos (2014), a division to three subsystems is implemented, chronologically:

- Aspect term labeling,
- Aspect-sentiment classification,
- Aspect-sentiment aggregation.

Aspect term labeling deals with extracting most prominent domain-specific single or multi-term aspects (e.g., “professor", “lecture quality", “literature") from a given text. It is important to keep in mind that for an aspect to be discussed in a given text, there does not necessary need to be an explicit aspect-word occurrence, or even a aspect-word synonym. For example, let‘s consider the following sentence:

“I liked the way he kept his pace while explaining during lectures.”

In the context of a student course questionnaire feedback, it would be safe to assume the aspect to be “Professor", although no explicit word occurrence of “professor", “lecturer" or “teacher" was mentioned. Aspect sentiment classification estimates the sentiment for every text referencing the target aspect term with three classes, e.g., *“Positive"*, *“Neutral"*, and *“Negative"*. Aspect-sentiment aggregation summarizes the overall sentiment towards an aspect.

2.3. Current state-of-art

Sentiment analysis can be divided by textual view into three categories (Collomb et al., 2014):

- Word / aspect level,
- Sentence level,
- Document level.

The amount of granularity can greatly affect the outcome of a model. Giving a document that is classified to either class, we lose information about which parts of the document are positive or negative. Sentence level granularity provides a better overview of the sentiment but it is not unusual for reviews and comments to have multiple sentiments addressing multiple aspects in the same sentence, therefore aspect level analysis provides the best level of textual granularity to be used.

Current state of sentiment analysis can be categorized into four methodological approaches:

– Lexical,
– Statistical,
– Machine learning,
– Deep learning.

While lexical and statistical approaches tend to be used primarily as key aspects for unsupervised learning, as in the recent work of (Fei et al., 2016), machine learning and deep learning approaches tend to perform better on most tasks (given enough labeled training data).

Detailed example of a lexical-based algorithm used in the work of (Hu i Liu, 2004) relies heavily on previous knowledge in the form of POS taggers, as it initially extracts distinct nouns and noun phrases, without determiners, for each review. The used algorithm then constructs an aspect term candidate list. For each candidate pair in a review sentence a new aspect term candidate is constructed. The candidate list is later sorted by the p-support value. The p-support value is the number of occurrences a term candidate occurs in sentences. What follows is a pruning stage with removal of non-compact aspect terms. Finally, a set of adjectives for each aspect term candidate is formed from sentences in which the term is mentioned. Candidates without corresponding adjectives are removed, the rest are sorted by the p-support value and returned as the solution. Variation on the proposed algorithm with improved results is discussed in Pavlopoulos (2014) with an added pruning stage using the popular Word2Vec model (Mikolov et al. (2013a), Mikolov et al. (2013b), Mikolov et al. (2013c)).

Most notable machine learning approach for building a sentiment classifier is the feature-based Support Vector Machine (SVM) (Pang et al., 2002), implemented by (Jiang et al., 2011). The authors manually designs all the target-independent features, thus proving for the model, to be an effective tool for coping with the given problem performed on small dataset. Nevertheless, manual feature-based models are still labor intensive and prone to subjective mistakes as well as lack of model interpetability.

Recent work in sentiment classification with deep learning methods includes classification with deep memory networks (Tang et al., 2015), classification with convolutional neural networks (Kim, 2014) and, more recently, sentiment classification using Long-short term memory networks (LSTM). Deep learning methods, providing enough training data, currently hold the state-of-the-art and were made possible with the recent breakthrough in word embeddings, primarily Mikolov's Word2Vector vector mapping (Mikolov et al., 2013b).

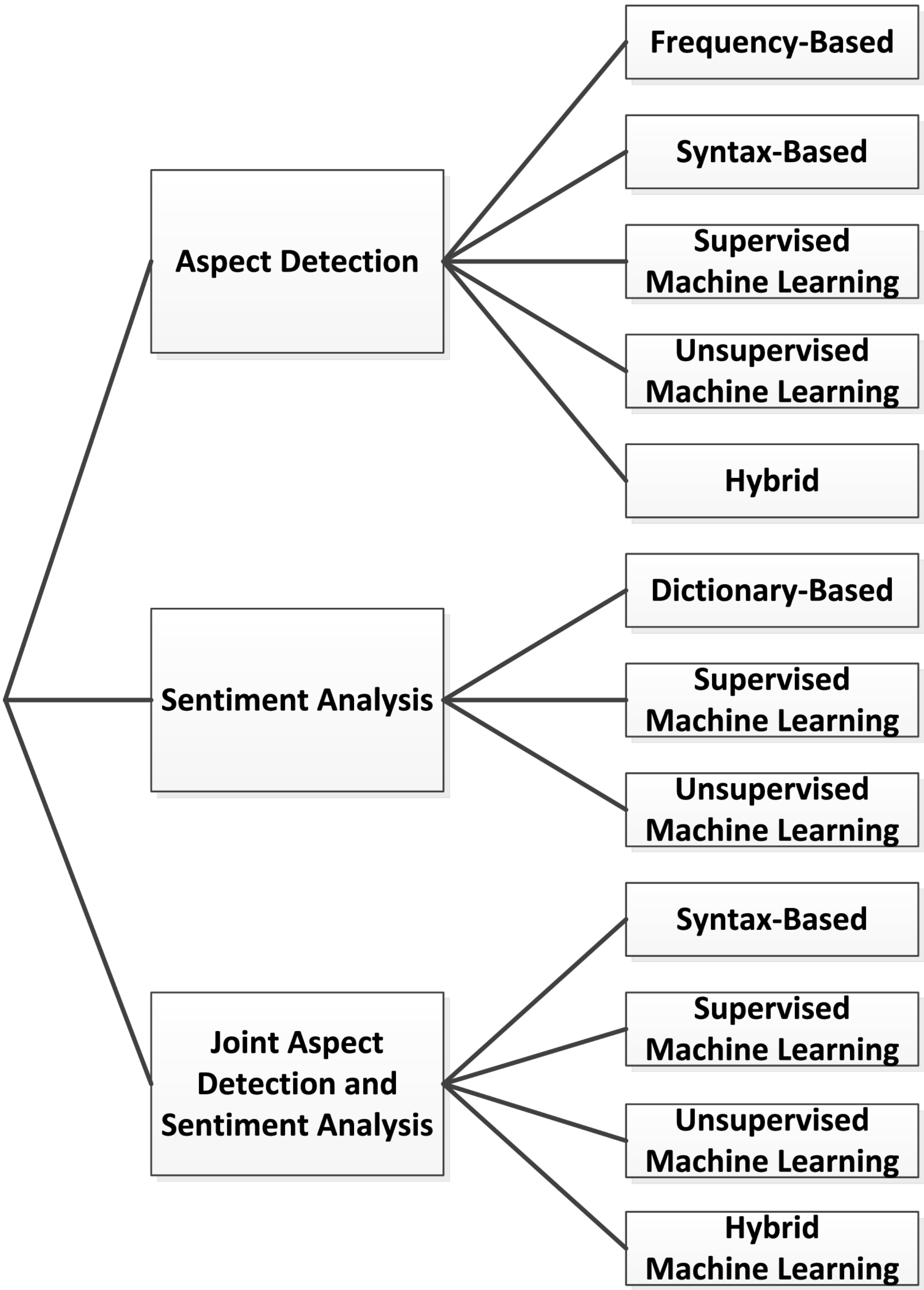

Figure 2.1: Taxonomy for aspect-level sentiment analysis approaches using the main characteristic of the proposed algorithm. Figure taken from Schouten i Frasincar (2016).

3. Datasets, Preprocessing and Word Embeddings

3.1. Dataset

In order to train the machine learning algorithms, later discussed in Chapter 4, a dataset is constructed. All examples are obtained from real student course questionnaires gathered by prof. dr. sc. Jan Šnajder at the Faculty of Electrical Engineering and Computing, University of Zagreb. This section provides an high-level overview of the overall dataset, as well as the performed proprocessing and labeling steps.

3.1.1. Initial Dataset Overview

The dataset consists of 458 student course questionnaires collected from four different courses from academic years from 2011 until 2017.[1] Mean number of words for each review, after preprocessing – which is discussed later in this section, is 122,8 with standard deviation of 24767,8. There are 56843 unique words used troughout the dataset. Student reviews vary in length, grammatical correctness, dialect, sentiment, and could be considered as highly noisy or versatile in terms of mentioned features. An example of a short, concise student course review that directly touches on aspects with a clear sentiment would be:

> *“Odličan predavač, nemam nikakve zamijerke :)"*

A more verbose student review that touches on multiple aspects with mixed sentiments, using multiple languages (primarily words that are Anglicisms), is the following:

[1] Those courses were: machine learning, artificial intelligence, programming in Haskell, text analysis and retrieval. All from the standard *curriculum* of the Faculty of Electrical engineering and Computing, University of Zagreb.

“Budući da Vi volite ankete i actually radite nešto s njima, a ne da samo zauzimaju mjesta u bazi, evo jedne.
Što se predavača tiče, nemam nešto posebno napisat: po običaju predavanja kvalitetna, ne može puno bolje. Materijali za učenje pripremljeni, za taj dio općenito svaka pohvala.
Nažalost, ispitna su vremena, nemam baš vremena bit opširan, ali evo da Vam malo pijem krv - zašto volim Machine Learning na Stanfordu, a odbojan mi je na FER-u. Evo jedan citat s predavanja prof. Andrewa Ng-a, kojim je to dobro sažeo.
"It turns out one of the other things we'll spend a lot of time on in this class is practical advice for applying learning algorithms. This is something that I feel pretty strongly about and it's actually something that I don't know if any other University teaches. Teaching about learning algorithms is like giving a set of tools, and equally important or more important than giving you the tools is to teach you how apply these tools.
I like to make an analogy to learning to become a carpenter. Imagine that someone is teaching you how to be a carpenter, and they say here's a hammer, here's a screwdriver, here's a saw, good luck. Well that's no good, right? You have all these tools but the more important thing, is to learn how to use these tools properly. There's a huge difference between people that know how to use these machine, learning algorithms versus people that don't know how to use these tools well.
Here in Silicon Valley where I live, when I go visit different companies, even the top Silicon Valley companies Very often I see people trying to apply machine learning algorithms to some problem, and sometimes they'll have been going at it for six months, but sometimes when they look at what they're doing, I say, you know, I could of told them like "Gee, I could of told you six months ago that you should be taking a learning algorithm and applying it in like the slightly motified way and your chance of success would have been much higher."
So what we're gonna do in this class is actually spend a lot of time talking about how, if you're actually trying to develop a machine learning system, how to make those best practices type decisions, about the way in which you build your system, so that when you're applying learning algorithm, you are less likely to end up one of those people who end up persuing some path for six months that you know, someone else could have figured out just wasn't gonna work at all and was just a waste of time for six months."
Ja se ponekad ovako osjećam na SU. Strogo formalizam svega i svačega, izvodi analitičkih rješenja, teorija vjerojatnosti... čemu sve to? Da li to ikome treba kad ide raditi neki algoritam strojnog učenja? Hoće li to nekome pomoći da napravi bolji algoritam strojnog učenja nego da to ne zna? Možda i hoće, ali za to je prvo potrebno da

shvati ideju algoritma, gdje i kako bi ga mogao koristiti, ali od ovakve šume detalja teško je to uvidjeti. Sve u svemu, strojno učenje je vrlo zanimljiva grana i ovaj predmet bi bio puno zanimljiviji i korisniji da je drugačije koncipiran. Ovako je jedan od onih predmeta koji su teški za položiti, a nema neke prevelike praktične vrijednosti, a zadovoljstva učiti pogotovo. "

Although rare, within student reviews there are occasional student reviews written mostly on a dialect instead of the standardized Croatian language:

"Predmet mi se čini OK, tema je zanimljiva, teorijska obrada je dovoljna za potrebe, eventualno da je morti malo više prakse i da su pitanja v zadaći bolje i jasnije napisana, puno stvari nije jasno (naravno, postoje konzultacije, al još je jednostavnije ak je odma sve jasno :D), inače, zadaća je zanimljiva, da se svašta nafčiti dok je se dela i drago mi je za to, žal mi je malo vremena jer su ispiti, da s i živce zgubiti doduše z njom, ostalo je sve OK, eventualno na ploči da malo lepše pišete, neka slova se teško čitaju, to je peh, a nije preveć kritično da bi se upozoravalo (uvek neko drugi vidi pa se od njega prepiše). Super su i one bilježnice z simulacijama, nikad ih nisam otprl, al mi se sviđaju dok se na satu pokažu, jedino kaj kradu vreme nekad dok je sat već gotovi, al vredi. JEDNA STVAR, DANAS SAM SLUŠAL ŽALOPOJKE I ARGUMENTE DA JE REGRESS V MATLABU GLUP I BLESAV ZA ZADAĆU I DA JE GLUPO KAJ NISMO DELALI GRADIJENTNI SPUST JER SE JEDNOJ OSOBI TO JAKO DOPADA JER JE NA TEČAJU ČULA ZA TO, AK SAM JA DOBRO SHVATIL, OSOBA JE ZVREĐALA SU (OVO JE MORTI V AFEKTU) I NE RAZME ZAKEJ NISMO DELALI GRADIJENTNI SPUST, AJME, D E L A L I S M O G A N A U Z O R K I M A, AL DOTIČNOJ OSOBI TO NIJE DOST, A OSOBNO MISLIM DA NIJE NI PROČITALA KAJ SMO TAM DELALI, FAKAT JADNO, BILO BI SUPER NEKAJ O TOJ METODI ONDA I OVDE MORTI REĆI, al bilo je teže o tome pričati, uglavnom, meni se ne čini da su za ove primjene trebale takve metode, ovo se lepo da z jednostavnim stvarima precizno dobiti, a ono nije tolko precizno i verojatno se koristi samo za gnjusobe od funkciji, BILO BI SUPER TO MORTI SPOMENUTI NEGDE AKO VEĆ NEMO DELALI ili tak nekaj, ovo danas je bilo vređanje SU-ja i još par stvari, tužno :P Odlično je isto kaj se puno spominju i engleski pojmovi tak da se vidi kak se to i drugde veli i koristi. Jedna primjedba, profesor Ribarić koji koristi slajdove za grafoskop (!!!) i koji je stariji je na uvodnom satu svojeg predmeta rekel kak se Raspoznavanje uzorki veli na engleskom, NJEMAČKOM i francuskom, na SU-ju di koristimo i prezentacije i Mathematicine bilježnice i Ubuntu i Windowse smo rekli samo za engleski :D Zbog verojatnosti čitanja

komentara i zbog odsustva slobode izražavanja, neke stvari radije ne bi komentiral, al predmet je sve skup super :D"

From the examples provided above it is evident that the dataset is extremely diverse in terms of grammatical rules used for writing as well as other textual features. Coupled with a relative small number of student course reviews, the dataset proves to be a suboptimal fit for standard domain algorithms, as discussed in Chapter 4.

3.1.2. Preprocessing Steps

A number of preprocessing steps are applied on student course questionnaires for all models evaluated in this paper. Each student course questionnaire is preprocessed with the following steps, chronologically:

1. Lowercased to combat mixed usage of Croatian diacritic signs within reviews;
2. Multiple occurrences of all special (non numerical or alphabetical) characters are reduced to a single occurrence of that character (e.g., "..." is transformed to "."');
3. Each review is split into word tokens – by splitting the text on newlines, whitespaces, and tabulators;
4. Stop-words are filtered from reviews. A list of used stop-words is available in Table 3.1 .

je da se u na ne to a za sam su mi ali od nije ja pa s sto bi koji ti ako sve samo ili ima kao jer iz o sa kad ce kako tako me po sad li bilo meni si znam mislim biti onda ovo nisam ga ni te bio malo do nema al netko ljudi nego bila koje koja jel tu nesto tu neki sto mogu koliko ih dobro taj prije ona puno jedan on nisu tko hvala svi treba zna kod ovaj ono nakon isto m ko ce toga bez jako bih oni sta mu opet zbog smo no gdje mene joj bit neke e jos zato par kada neka kaj uz čak što možda niti vise

Table 3.1: Stopword list obtained from the Word2Vec model by retrieving most frequent words.

3.1.3. Labeling the Dataset

An important decision prior to the step of data labeling is deciding the granularity over which to split the text data. This domain specific problem provides the following options:

1. Sentence level split,
2. Paragraph level split,
3. Document level split,
4. Arbitrarily split (e.g., from sentence to paragraph).

Let's discuss each approach in greather depth. Splitting the dataset on sentences divides the student course questionnaires into a granularity that is consistent throughout the train and test times by always splitting student course reviews over interpunction characters. Although this is beneficial in terms of implementation, this method suffers from aspect sentiment fragmentation as an aspect could be discussed in multiple sentences throughout a student course review – therefore an additional aggregation step is required to be performed over the overall evaluation system to capture the aggregated sentiment for each discussed aspect.

Paragraph level split is viable under some heavy constraints. Foremost, highly structured review text with each aspect split into separate paragraphs and low variability of paragraph lengths. As described in this chapter, this approach would not suffice as the dataset has proven to be variable in length and highly unstructured.

No splitting performed, or document level split, is also not viable due to high variability of number of sentences in student reviews.

Finally, arbitrarily splitting the dataset would probably yield in a most information gain for the model, but given a high-quality arbitrarily split labeled data, how would we go detecting aspect-sentiment occurrences at evaluation time? This approach would require constructing an additional step of aspect detection in the system, as well as for aggregation.

In light of this new information, a sentence level split is introduced and performed. All preprocessing steps were applied, as previously discussed in this Chapter, with an addition of sentence granularity level splitting step between the multiple character removal step and the tokenization step (steps two and three, respectively). Therefore, every student course questionnaire is split on question marks, periods, exclamation marks, etc. to form a number of student course review sentences.

The newly split dataset was labeled with ten human-chosen, domain-specific aspects. Those aspects are:

- Predmet (engl. *Subject*),
- Profesor (engl. *Professor*),
- Literatura (engl. *Literature*),
- Materiali (engl. *Materials*),
- Ispiti (engl. *Exams*),
- Predavanja (engl. *Lectures*),
- Domace zadace (engl. *Homeworks*),
- Asistenti (engl. *Assistants*),
- Gradivo (engl. *Content*),
- Nista od navedenoga (engl. *None-of-the-above*).

For each aspect on a given sentence, a sentiment label was assigned. Labeling was performed with three sentiment classes – “Positive", “Neutral", and “Negative". The annotation was made by the author of this thesis. An aspect was labeled for a given sentence if that sentence contained any word, or word synonym, from the predefined aspect group. In other words, if a sentence contains an aspect word, it was labeled as a member of that aspect. The data labeling took approximately 20 hours to reach the current number of labeled examples and was performed in two consecutive days. The resulting dataset consists of 739 labeled sentences. A couple of examples are provided as follows:

“snajder jako dobro predaje potice diskusiju sto je izrazito vidljivo u engleskoj (manjoj) grupi" – (Professor, Positive)

“domaće zadaće trebale su zbilja doći ranije" – (Homeworks, Negative)

“sve pohvale predavaču i predmetu" – (Professor, Positive); (Subject, Positive)

The final labeled dataset percentage, in terms of aspect and sentiment statistics, is seen in Table 3.2 and Table 3.3 . From the provided tables it is visible that single-aspect student course review sentences dominate the dataset with a high percentage of “Neutral" sentiments. The provided dataset statistics prove to be beneficial as they enable effective one-aspect sentiment training for machine learning models.

	1	2	> 2
Number of sentences	631	98	10
Overall %	85	14	1

Table 3.2: Breakdown of the number of sentences per number of aspects.

	Positive	Neutral	Negative
Sentence count	127	377	127
Overall %	20	60	20

Table 3.3: Sentiment split among single aspect labeled sentences.

3.2. Word Embeddings

Through recent years, a popular word embedding model is used as a primary method of mapping words into a single vector space – Word2Vec (Mikolov et al. (2013a), Mikolov et al. (2013b)). Previously, most papers focused on a bag-of-words approaches (BOW), which assigns an unique id number to each word in a text corpus. Such systems would then represent documents as a vector of word id's used in those same sentences. Although intuitive, bag-of-words vectors do not provide any additional information between those vectors – it is upon the model to learn the abstractions between those representations. Word2Vec, on the other hand, provides word similarity information within the embedding model and can be successfully used for tasks like finding word synonyms.

Word2Vec, in its essence, is a two-layer artificial neural network that maps text corpus to a set of N dimensional vectors. For the problem of aspect retrieval, and later sentiment classification, a Word2Vec model is trained on a topic specific dataset on Croatian language.

3.3. Word2Vec Model Construction

Source for text data used for training the Word2Vec model was scraped from a popular Croatian community-driven forum website.[2] The community driven and unformally written text seems to follow the domain specific text structure of students that wrote course questionnaires.

Due to overwhelming amount of threads on the forum, only four were used. For ant thread, all text from sub-threads was extracted and saved for training. Those threads were:

- Fakulteti i visoka učilišta (engl. *Faculties and colleges*),
- Glazbenici (engl. *Musicians*),
- Film (engl. *Movie*),
- Televizija (engl. *Television*).

The reasoning behind the selected threads was that those threads (e.g., “Glazbenici", “Film", “Televizija") will contain semantic, subjective comparative comments close related to the problem domain and resemble the student course questionnaire dataset. The constructed dataset contains about 960K forum posts and is 3.8 GB in raw, uncompressed size. Post-replies were not filtered from posts so there is a possibility of text duplication which in turn could affect the Word2Vec trained model. Text repetition effects could be considered negligible as a model trained on unstructured, misspelled text is desired to mimic the sentence constructs of students in course questionnaires.

The dataset was preprocessed as follows:

- Due to the free grammatical policy found on todays community driven websites like forum.hr, some words tend to have Croatian diacritical characters and others don't. This subtle distinction could greatly influence training of the Word2Vec model as words like “želja" and “zelja" would be treated as distinct although they are the same in terms of word meaning. Therefore, removal of Croatian characters carrying diacritical symbols is performed. Primarily replacing characters like č, ć, đ, š, ž with c, d, s, z, respectively;
- filtering and removal of all special characters except letters, numbers and whitespaces.

With the dataset in-place, training of the Word2Vec model was in turn. In order to train the word embedding model an popular topic modeling framework was used –

[2] `http://www.forum.hr`

Gensim.[3]

The final Word2Vec model is 732 MB in size while representing 56843 words. The constructed vectors are 300 dimensional and the Word2Vec training model variant is CBOW. No stemming or lemmatization steps were performed.

[3] http://radimrehurek.com/gensim/

4. Models and Evaluations

In this Chapter various unsupervised and supervised models are constructed, evaluated and discussed. Chapter is split into three main system sections: aspect detection, aspect-sentiment classification, and aspect-sentiment aggregation. Each section offers its insights, conclusions, and discussion regarding the relative subject.

4.1. Aspect Detection Models

This section talks about aspect extraction and classification results obtained from the dataset split described Chapter 3. The problem is formulated as follows. For each new student course review document, label all aspect that directly or indirectly occur in the given document. Besides a supervised SVM model, an LDA approach was evaluated with the KL-divergence measure to identify the correct number of topics. Finally, an unsupervised method that utilizes Word2Vector word embedding properties, most notably successfully localizing words that are synonyms, to extract aspect synonyms into aspect groups that were after on used for detection and extraction.

4.1.1. SVM Model

For classification, a non-linear SVM (Support vector machine) implementation is used with default parameters[1] and a *rbf* (Radial basis function) kernel. Implementation for the SVM is part of the Scikit-learn[2] package.
The Word2Vec discussed in Chapter 3 is now utilized for mapping words to their 300 dimensional vectors. Additive composition method is applied, by which word-vectors of sentences are summed to form a single vector representing a single review sentence. Those sentence vectors form the input data for the SVM classifier.

[1] `http://scikit-learn.org/stable/modules/svm.html`
[2] `http://scikit-learn.org/`

Cross-validation was performed with a 5-fold split. The results was a trained model with an 12 % accuracy score with a 4 % deviation.

4.1.2. Latent Dirichlet Allocation (LDA) Model

An unsupervised approach based on Latent Dirichlet Allocation (LDA) (Blei et al. (2003)) was used for aspect-topic extraction (Mei et al. (2007), Jo i Oh (2011)).
As stated by Pavlopoulos (2014), LDA model assumes that document d, which consists of $w_1, w_2, ..., w_{\mathrm{m}}$ words (where $m = |d|$), are formed by continuously selecting a topic t from a multinomial probability distribution $P(t|d)$ over T specific topics for each document.[3]
To test the applicability of this model to the provided dataset, an unsupervised metric is used for selecting the optimal number of topics, called Kullback–Leibler divergence (KL divergence). This metric, described in Arun et al. (2010), is a measure of how one probability distribution diverges from a second – expected probability distribution and on its own is not a distance measure but a measure of entropy. The optimum number of topics k is then selected by choosing a value for which the KL divergence suddenly drops for multiple runs on the same dataset.

Evaluation

In Figure 4.1 we see the results obtained from running the LDA model on the provided dataset. The three preformed executive runs, with number of topics ranging from 0 to 18, provided inconsistent KL divergence values which ranged widely. We see an absence of consistency and no sudden drops which could imply an optimal number of topics for the LDA model.

Overall, LDA has proven to be a inadequate model for this dataset, for which may be due to:

- small dataset size,
- dataset inconsistency of topics,
- noise in the provided dataset.

[3]Assuming that documents are created by choosing some fixes number of topics in a particular mixture.

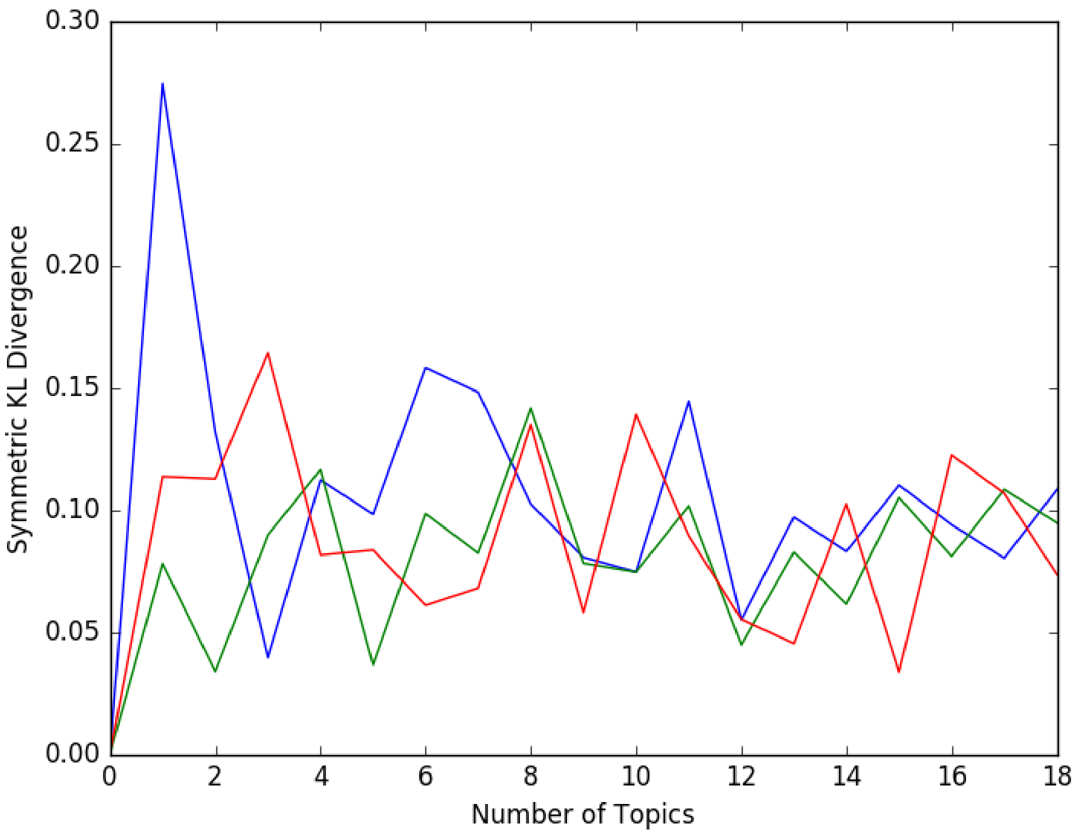

Figure 4.1: Kullback–Leibler divergence values for LDA model fitting with number of topics ranging from 0 to 18 on 3 distinct runs (blue, red, green).

4.1.3. Unsupervised Aspect-Term Matching

Results obtained from the supervised SVM approach and the unsupervised LDA approach had provided a valuable feedback in terms of amounting the dataset noisiness and fragmentation. To recapitulate, our supervised approach provided an accuracy of 12 % with a standard deviation of 4 %. The LDA approach failed to converge around a consistent topic number integer – according to the KL-divergence measure.

An arguably straightforward aspect-term matching method is implemented that utilities two important features:

- Target aspects are predefined and known prior working with the dataset,
- There is available a previously trained Word2Vector[4] model trained specifically, or at least marginally, within the target domain.

For the list of predefined aspects, neighborhood words are extracted from the Word2Vec

[4]The reason why Word2Vector performs well is that it successfully captures words to word synonyms in its resulting vectors space, therefore neighborhood of a word will consists of words that are used interchangeably.

model for each aspect – forming aspects word groups. Neighborhood is in this context defined as N neighboring words for which the corresponding MAP score is highest. By empirical evaluation on the dataset, N is chosen to be four. Concretely on this problem, the resulting groups are visible in Table 4.1.

	Neighboring word-vectors
Predmet	kolegiji, ispit, smjer, seminar
Profesor	asistent, prof, predavac, profa
Literature	skripta, knjiga, katedra, skriptica
Materijali	ispiti, kolegiji, primjeri, zadaci
Ispiti	kolokviji, predmeti, rokovi, kolegiji
Predavanja	vjezbe, konzultacija, predavanje, seminare
Zadace	vjezbe, zadace, vjezbama, vjezbi
Asistenti	profesori, predavaci, demosi, kolokviji
Gradivo	gradiva, ucenje, znanje, ponavljanje

Table 4.1: Word2Vec Aspect-word neighborhood.

From the resulting aspect-neighborhood groups we eliminate neighbors that are lemmas of other aspects. So, for instance, we remove “Ispiti" from the aspect-neighborhood group “Materijali" and similar. An word lookup is performed using lemmatized aspect-neighborhood words, mapping each student review sentence to its corresponding aspect-group. This method proved to be a efficient way of labeling aspects for given comments as its resulting Mean average precision (MAP) score is 0.591 on the labeled student questionnaire sentence dataset.[5]

4.2. Aspect-Sentiment Classification

Text classification by sentiment polarity is a popular research subject where large parts of previous work was concentrated around assigning labels (e.g., positive, neutral, negative) to various levels of text granularity (e.g., token, sentence, paragraph, document).

[5]Calculated using Sci-kit Average precision score implementation.

Aspect based sentiment analysis (ABSA) specifically targets determining sentiment polarity for every aspect term mentioned within a document. This task, of assigning sentiment to each paragraph, can be considered extremely challenging as adjectives, assigned to nouns, can have different sentiment interpretation depending on the context and the noun they modify. For instance, in the sentences:

“He was quick on the trail."

“It was not until he submitted his exam, he realized his error - he was quick minded."

Word “quick", as a single token, modifies the aspect sentiment considerably.

4.2.1. Sentence-level Dataset Split

The used labeled dataset is as described in Chapter 3, which consists of 739 labeled student course questionnaire sentences. From Table 3.2 we can see than out of 739 labeled sentences 631, or 85 %, contain only a single aspect. For those sentences we can determine the sentiment which the whole sentence carries – therefore to train the classifier just single-aspect sentences are used. The training data consists of 631 sentences with sentiment distribution described in Table 3.3. Dataset split seems prominently dominated with neutral sentences with exact amount of positive and negative sentiment counts.

	with stop-word removal	without stop-word removal
Accuracy	0.52±0.10	0.62±0.02

Table 4.2: Training results for a single level classifier regarding stop-word removal.[6]

[7]All evaluations were performed using 10-fold cross validation.

	with stop-word removal	without stop-word removal
First level SVM accuracy	0.63 ± 0.02	0.63 ± 0.03
Second level SVM accuracy	0.55 ± 0.08	0.54 ± 0.06

Table 4.3: Training results for a two level classifier regarding stop-word removal.[7]

4.2.2. Sentence-level SVM Sentiment Classification

A popular approach for classifying sentiment in short texts (e.g., sentences, tweets, etc.) consists of two levels of classifiers. The first level classifies each given message if it is objective[8] or subjective.[9] The second level of classification classifies only sentences that are not objective, i.e., only sentences that carry sentiment, and its classification goal is to determine if the sentiment if positive or negative. This system of sentiment classification, using two classifiers as seen in Figure 4.2 using two classifiers, has two important advantages:

- Proposed two-level decomposition allows us to act upon class imbalance of the dataset we work with if we have more neutral messages than negative or positive;
- Modularity of the system allows easy subjectivity information retrieval of given text where that is necessary.

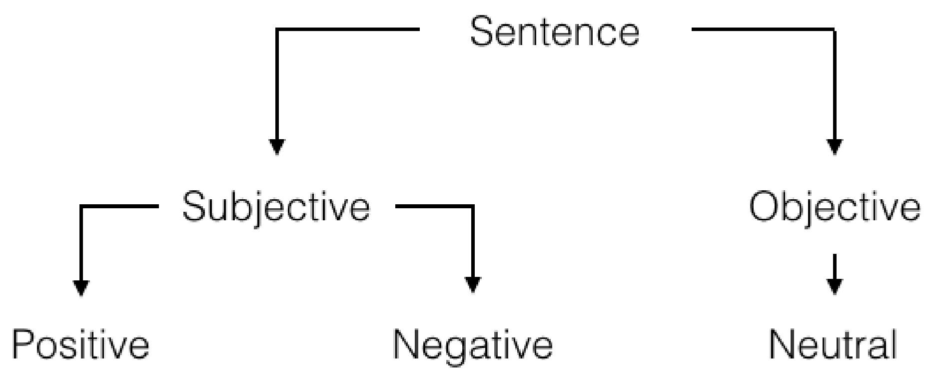

Figure 4.2: Two level objectivity, subjectivity classifier model representation.

[8] Document does not carry positive or negative sentence.
[9] Document carries some form of positive and/or negative sentiment.

N-gramming Sentiment Classification

To amplify the effects of adjectives modifying the noun (assuming they carry majority of the sentiment) an n-gramming data preprocessing step is introduced to input sentences to isolate neighboring words of a aspect. Concretely, results performing n-gramming with a windows size of five performs best. Trough the following evaluation the two neighbouring words to the left and right, respective to the aspect word, are used for classification.

In the problem implementation the described principle was used with two SVM classifiers and later on compared with a system that used single level SVM sentiment classification. Both systems were trained to assign 3-way polarity estimates (i.e., positive, neutral, negative) and their training results are shown in Table 4.2 and Table 4.3. Interesting to observe are the training results for both single-level classification and two-level classification regarding how stop-word removal greatly affects the results – implying stop-words do carry sentiment and, in this case, prove to be of great importance as the training accuracy improves by 10 %. Moreover, the two-level classification system provides a detailed insight into the same effect – from the training data we observe how the training accuracy of subjective-objective sentences is not greatly affected with stop-word removal, but positive-negative sentence classification, to some extent, is. Although in most Natural language processing (NLP) tasks stop-word removal is a common step, as it is reasoned that stop-words carry no information in a sentence, it seems not all systems require such reasoning and each tasks should be approached individually.

Both systems were tested on a test dataset that consisted of 25 % of the overall dataset. The following metrics were evaluated: Precision, Recall, F1 score, Support. Evaluation results for the single-level SVM classifier are in Table 4.4, while results for the two-level system consisting of two SVM classifier are in Table 4.5 and Table 4.6. All SVM's in evaluation used a linear kernel and with hyper-parameter C = 1, together with all other parameters set to default values as provided in the Sci-kit implementation of SVM.[10]

[10] `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC`

	Precision	Recall	F1 score	Support	Baseline accuracy %
Negative	0.67	0.06	0.11	33	21
Neutral	0.62	0.99	0.76	93	59
Positive	0.88	0.22	0.35	32	20

Table 4.4: Test results for a single layer SVM classifier without stop-word removal, tested on 25 % of the dataset.

	Precision	Recall	F1 score	Support	Baseline accuracy %
Objective	0.64	1.0	0.77	93	58
Subjective	1.0	0.14	0.25	65	42

Table 4.5: Test results for a two layer SVM classifier (first level) without stop-word removal, tested on 25 % of the dataset.

	Precision	Recall	F1 score	Support	Baseline accuracy %
Negative	0.55	0.94	0.69	35	53
Positive	0.67	0.13	0.22	30	47

Table 4.6: Test results for a two layer SVM classifier (second level) without stop-word removal, tested on 25 % of the dataset.

4.3. Aspect-sentiment Aggregation

This section focuses on aggregating aspect-sentiment pairs. To avoid multiple sentiment scores for same aspects we aggregate them, thus providing one sentiment score per aspect. Aspect-sentiment aggregation in the implemented system occurs after aspect extraction and sentiment assignment to aspects. In a sentence that contains multiple occurrences of the same aspect, after the two previously mentioned steps, an aspect can have multiple sentiments assigned to it e.g., Table 4.7

One possible way would be to average sentiment values for each aspect, while coding each sentiment with a number. For instance we could consider positive to be 1.0, neutral 0.0, and negative -1.0 . Averaging the provided example we get the following distribution visible in Table 4.8 .

From Table 4.8 It is evident that we cannot assign to the “Professor" aspect either the “Positive" or the “Neutral" label. We could for instance round to whole numbers,

Aspect	Sentiment
Professor	Positive
Professor	Neutral
Lectures	Negative
Assistant	Neutral

Table 4.7: Possible sentiment assignments to aspects.

Aspect	Sentiment
Professor	0.5
Lectures	-1.0
Assistant	0.0

Table 4.8: Possible numerical sentiment assignments to aspects.

but that would introduce unnecessary inaccuracy.
Therefore the final system takes into account the real number classification scores and assigns the aspect an average score then maps it to a referent sentiment label.

5. Web Application

For easier usage of the previously described evaluation system, a web application was made, as shown in Figure 5.1 . From the provided screenshoot it is visible that the overall web application is divided intro two distinct parts:

1. Student text review area,
2. Aspect define area.

This Chapter focuses on explaining the mechanics of the web application and the technologies it uses.

5.1. Web Application Overview

Application usage is expected as follows. Initially the user inputs the student review text into the *Student text review area*. The input text can be unstructured, paragraphed, with or without interpunction etc. Then, a set of aspects is repeatedly inputted into the *Insert aspect* input filed until all aspects of interest are covered. For more convenient use of the web application, a predefined list of aspects that were originally used throughout the paper is used that can be edited freely. Finally, by pressing the *Evaluation* button in the *Student text review area*the text is send onto evaluation. The evaluation results are presented in the *Aspect define area* as aspect-sentiment pairs. For each aspect that is recognized within the text, a sentiment value of "Positive", "Neutral", and "Negative" is evaluated and presented in the web application.

An example run is provided in Figure 5.2. From the Figure, we can see the evaluation results for a randomly selected student review. This example run successfully captures the sentiments behind the review as most of the aspects which are mentioned in the review are indeed “Negative", while the overall aspect “Predmet" is considered, by the reviewer, “Positive". The evaluation assigns sentiments only for aspects it finds, therefore for the aspect “Asistent", sentiment is correctly left blank as no occurrence of that aspect were found in the review.

This evaluation run is an good example of the domain specificity of the problem set, as most reviews tend to primarily focus on negative aspects first, but on the whole, overall or global aspect is in fact considered positive. This observation implies that text granularity plays a vital role in forming a sentiment hierarchy toward various levels of aspects.

5.2. Frameworks and Technologies

Development of the web application was divided into two parts – frontend and backend.
For developing the fronted, a novel programming language was used – Elm-lang[1]. Elm is a functional programming language that cross-compiles to Javascript and HTML – providing faster execution speeds and no run-time exceptions.
The backend, implemented in Flask[2], consists of a single POST request declaration for handling a incoming JSON request containing the text, aspects and returning a evaluation result with sentiments assigned to appropriate aspects.

[1] http://elm-lang.org/
[2] http://flask.pocoo.org/

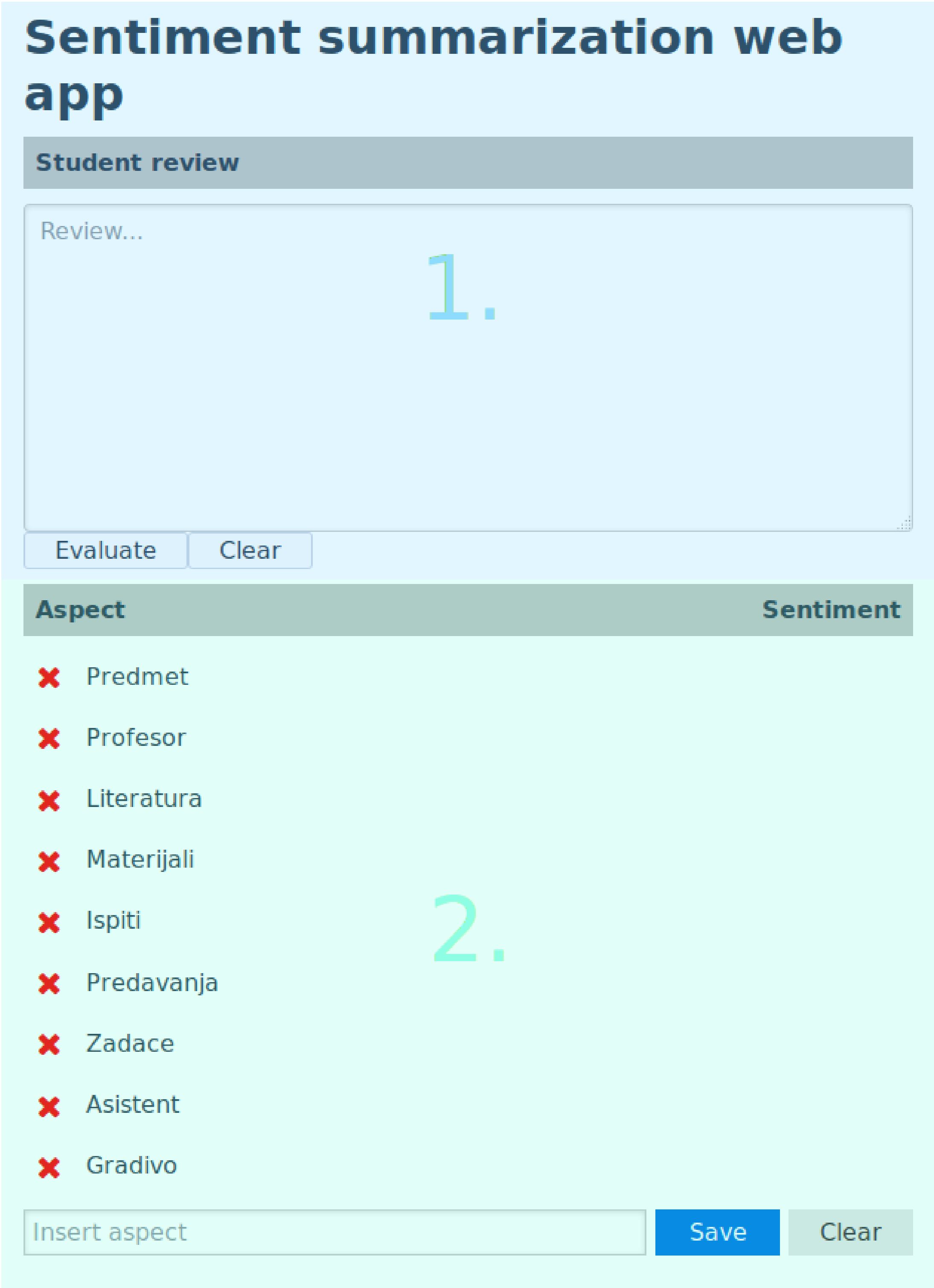

Figure 5.1: Initial state screenshoot of the evaluation system web application with colored regions.

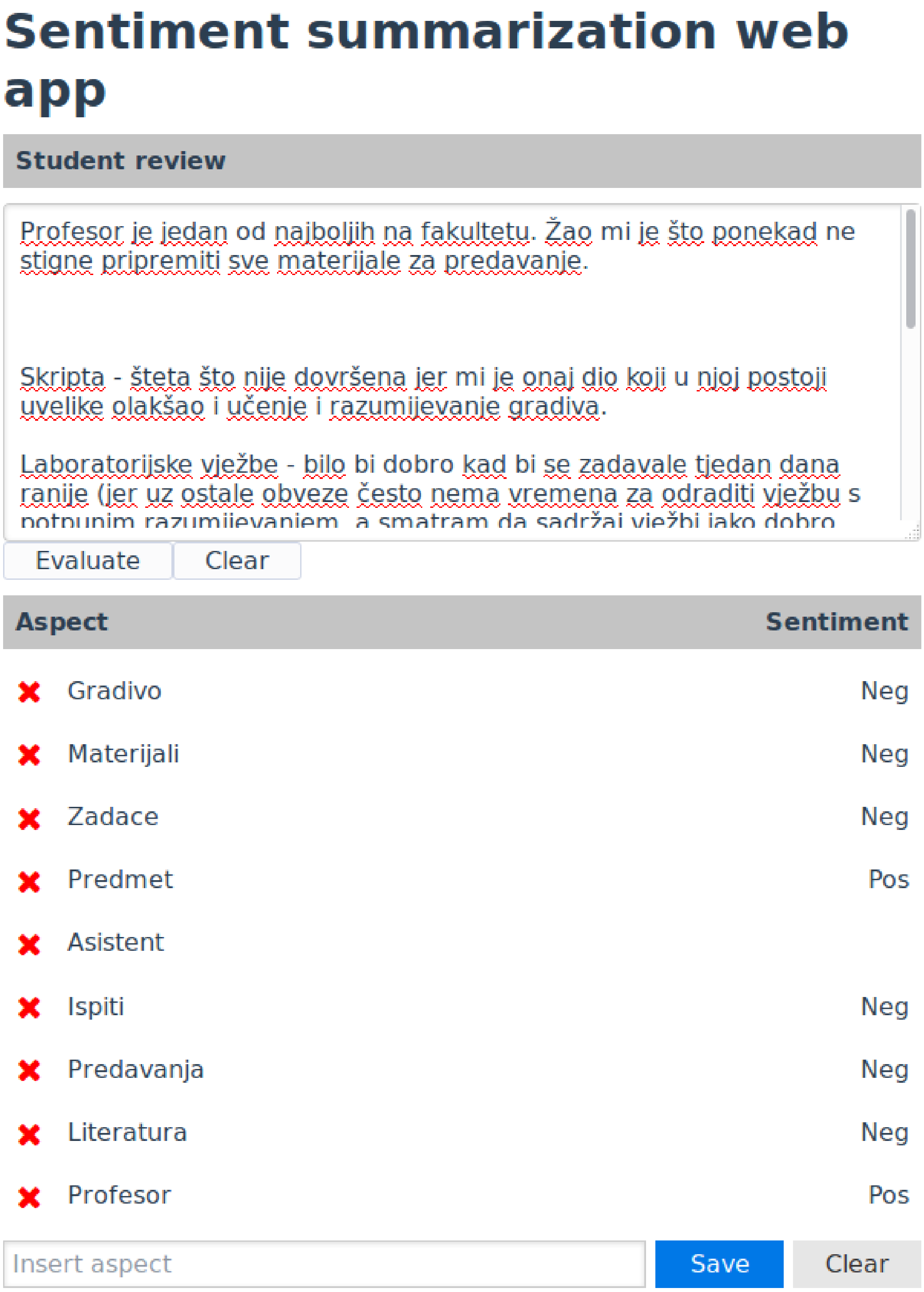

Figure 5.2: Screenshoot of an evaluated example in the web application.

6. Conclusion and Future Improvements

Subjective, sentiment filled feedback from students provides a valuable feedback to professors and in the same time proves to be an interesting dataset to explore. Currently, due to large number of student questionnaires being made every semester quarter, a potentially helpful high-level overview of the overall sentiment is missed, or obtained with time exhaustive reading by professors.

Natural language processing (NLP) is a scientific discipline resulting from three major overlapping fields of study: computer science, linguistics, and artificial intelligence. Recent breakthroughs in machine learning[1] provide new tools for modeling and reasoning about natural human language language.

This bachelor thesis implemented an aspect-based sentiment analysis system for student course questionnaires. Three system stages were individually evaluated and tests. Those tree stages are:

1. Aspect term labeling,
2. Aspect-sentiment classification,
3. Aspect-sentiment aggregation.

For aspect term labeling, the best performing model was a synonym grouping model over Word2Vec vectors, with an MAP score of 0.59 . For aspect sentiment classification best performed a n-gram based text model with a two level SVM classifier, with an accuracy of 63 % and 55 % respectively.

The ability to evaluate vast amounts of text over a common feature or aspect proves to be of great importance both for the industry and academia. In the domain for which this system was built, it could be useful for quickly evaluating large amounts of student

[1] Machine learning is a subfield of computer science.

review, hoping to find some sentiment consistencies. Due to the overall dataset diversity and size, as discussed in Chapter 1, limitations are introduced. Some of which are:

- Sentiment detection is supervised, therefore the performance of classification is heavily influenced by the quality of the training data;
- Multi word aspects are not supported, so the system operates only one single word aspects;
- Croatian localization and model training, and therefore can not be used for any other language;
- Aspect detection is not unsupervised.

By introducing a bigger (preferably, less noisier dataset) and more labeled data, this system could yield a greater improvement in performance with the current implemented algorithms. In the future work, an overall performance improvement is expected with the introduction of more unsupervised machine learning algorithms, like LDA – as proposed in Chapter 3.

Bibliography

R. Arun, V. Suresh, C. E. Veni Madhavan, i M. N. Narasimha Murthy. *On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations*, stranice 391–402. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-13657-3. doi: 10.1007/978-3-642-13657-3_43. URL `http://dx.doi.org/10.1007/978-3-642-13657-3_43`.

David M. Blei, Andrew Y. Ng, i Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Ožujak 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`.

Anaıs Collomb, Crina Costea, Damien Joyeux, Omar Hasan, i Lionel Brunie. A study and comparison of sentiment analysis methods for reputation evaluation. *Rapport de recherche RR-LIRIS-2014-002*, 2014.

Geli Fei, Zhiyuan Brett Chen, Arjun Mukherjee, i Bing Liu. Discovering correspondence of sentiment words and aspects. U *In proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, 2016.

Minqing Hu i Bing Liu. Mining and summarizing customer reviews. U *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, stranice 168–177. ACM, 2004.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, i Tiejun Zhao. Target-dependent twitter sentiment classification. U *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, stranice 151–160. Association for Computational Linguistics, 2011.

Yohan Jo i Alice H. Oh. Aspect and sentiment unification model for online review analysis. U *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, stranice 815–824, New York, NY, USA, 2011. ACM.

ISBN 978-1-4503-0493-1. doi: 10.1145/1935826.1935932. URL `http://doi.acm.org/10.1145/1935826.1935932`.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Bing Liu. Sentiment analysis and subjectivity. U *Handbook of Natural Language Processing, Second Edition*, stranice 627–666. Chapman and Hall/CRC, 2010.

Qiaozhu Mei, Xuehua Shen, i ChengXiang Zhai. Automatic labeling of multinomial topic models. U *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, stranice 490–499, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281246. URL `http://doi.acm.org/10.1145/1281192.1281246`.

Tomas Mikolov, Kai Chen, Greg Corrado, i Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, i Jeff Dean. Distributed representations of words and phrases and their compositionality. U *Advances in neural information processing systems*, stranice 3111–3119, 2013b.

Tomas Mikolov, Wen-tau Yih, i Geoffrey Zweig. Linguistic regularities in continuous space word representations. U *Hlt-naacl*, svezak 13, stranice 746–751, 2013c.

Bo Pang, Lillian Lee, i Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. U *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, stranice 79–86. Association for Computational Linguistics, 2002.

Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

Ioannis Pavlopoulos. Aspect based sentiment analysis. *Athens University of Economics and Business*, 2014.

Kim Schouten i Flavius Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2016.

Duyu Tang, Bing Qin, Xiaocheng Feng, i Ting Liu. Target-dependent sentiment classification with long short term memory. *CoRR, abs/1512.01100*, 2015.

Appendices

A. System Execution Instructions

To introduce maximum reproducibility of this bachelor thesis results and due to high numbed of package dependencies that were used in creating this project, a *dockerization*[1] process was introduced. This allows the system additional benefits, like:

- Minimal setup time on a new execution environment
- Future-proof system that already contains all the required packages and dependencies

Make sure you have the latest version of Docker[2] installed and fully working locally on the machine you are using. Then load the saved thesis docked image into Docker by using:

```
$ docker load -i <path_to_image_tar_file>
```

Then run an instance of the newly loaded image with:

```
$ docker run -p <port-number>:4000 <image_name>
```

By running this command you are mapping a <port-number> from your local machine to a port in the docker container. After a few moments you can access the system by entering "localhost:<port-number>" into your Internet browser. For instance, if the used <port-number> was 8000, you would access the system by typing in your browser:

```
localhost:8000
```

[1] Dockerizing an application is the process of converting an application to run within a Docker container.

[2] https://www.docker.com/

Sažimanje sentimenta u studentskim upitnicima predmeta

Sažetak

Sažimanje sentimenta jest zadatak obrade prirodnog jezika koji kombinira sažimanje teksta i analizu sentimenta. Osnovna ideja jest generirati sažetke subjektivnih tekstova koji u obzir uzimaju polaritet sentimenta prema pojedinim aspektima proizvoda ili usluge. Sustavi za zažimanje sentimenta omogućavaju analizu velikih količina subjektivnih tekstova, npr. korisničkih recenzija ili komentara na društvenim mrežama. Jedna zanimljiva i korisna primjena sažimanja sentimenta jest analiza studentskih upitnika o predmetima, kod kojih studenti iskazuju svoje mišljenje o pojedinačnim aspektima predmeta.

Tema završnoga rada jest sažimanje sentimenta iz tekstova studentskih upitnika na hrvatskome jeziku. U okviru završnoga rada proučeni su postupci za ekstraktivno sažimanje teksta, postupci za klasifikaciju polariteta sentimenta te postupci za sažimanje sentimenta temeljene na strojnom učenju. Osmišljen je i implementiran postupak za sažimanje studentskih upitnika koji može identificirati pojedine aspekte predmeta, grupirati komentare koji se odnose na slične aspekte te ih klasificirati prema sentimentu. Izrađen je prikladan skup podataka za treniranje i ispitivanje modela, temeljen na stvarnim studentskim upitnicima. Na prikladan i intuitivan način prikazani su rezultati sažimanja sentimenta. Provedeno je eksperimentalno vrednovanje postupka na ispitnim podatcima. Radu je priložen izvorni i izvršni kod razvijenog sustava, skup podataka i programska dokumentacija te citirana i korištena literatura.

Ključne riječi: Obrada prirodnog jezika, strojno učenje, analiza sentimenta, analiza sentimenta na osnovu aspekata, Hrvatski jezik, upitnici, recenzije

Sentiment Summarization from Student Course Questionnaires

Abstract

Sentiment summarization is a natural language processing task that combines document summarization and sentiment analysis. The main idea is to generate summaries of subjective texts that account for sentiments toward the different aspects of a product or service. Sentiment summarization systems enable the analysis of large amount of subjective texts, e.g., user reviews or comments on social networks. One interesting and useful application of sentiment summarization is the analysis of student course questionnaires.

The topic of this thesis is sentiment summarization from student course questionnaires in Croatian language. Do a literature survey on extractive document summarization, sentiment polarity classification, and sentiment summarization. Devise and implement a method for sentiment summarization from student course questionnaires that can identify the various aspects of a course, group comments pertaining to similar aspects, and classify their sentiment polarity. Compile a suitable dataset for training and evaluating the model, derived from real student questionnaires. Devise a suitable and intuitive way for visualizing the sentiment summarization results. Carry out an experimental evaluation of the method on test data. All references must be cited, and all source code, documentation, executables, and datasets must be provided with the thesis.

Keywords: Natural language processing, machine learning, sentiment analysis, aspect-based sentiment analysis, Croatian language, student questionnaires