UMEÅ UNIVERSITY

Natural Language Processing in Cross-Media Analysis

Yonas Demeke Woldemariam

Licentiate Thesis, May 2018
Department of Computing Science
Umeå University
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

yonasd@cs.umu.se

ISBN 978-91-7601-885-9
ISSN 0348-0542
UMINF 18.06

Printed by UmU Print Service, Umeå University, 2018

Abstract

A *cross-media analysis framework* is an integrated multi-modal platform where a media resource containing different types of data such as text, images, audio and video is analyzed with metadata extractors, working jointly to contextualize the media resource. It generally provides cross-media analysis and automatic annotation, meta data publication and storage, search and recommendation services. For on-line content providers, such services allow them to semantically enhance a media resource with the extracted metadata representing the hidden meanings and make it more efficiently searchable. Within the architecture of such frameworks, Natural Language Processing (NLP) infrastructures cover a substantial part. The NLP infrastructures include text analysis components such as parser, named entity extraction and linking, sentiment analysis and automatic speech recognition.

Since NLP tools and techniques are originally designed to operate in isolation, integrating them in cross-media frameworks and analyzing textual data extracted from multimedia sources is very challenging. Especially, the text extracted from audio-visual content lack linguistic features that potentially provide important clues for text analysis components. Thus, there is a need to develop various techniques to meet the requirements and design principles of the frameworks.

In our thesis, we explore developing various methods and models satisfying text and speech analysis requirements posed by cross-media analysis frameworks. The developed methods allow the frameworks to extract linguistic knowledge of various types and predict various information such as sentiment and competence. We also attempt to enhance the multilingualism of the frameworks by designing an analysis pipeline that includes speech recognition, transliteration and named entity recognition for *Amharic*, that also enables the accessibility of *Amharic* contents on the web more efficiently. The method can potentially be extended to support other under-resourced languages.

Preface

This thesis contains a brief description of natural language processing in the context of cross-media analysis frameworks and the following papers.

Paper I Yonas, Woldemariam. Sentiment Analysis in a Cross-Media Analysis Framework. *2016 IEEE International Conference on Big Data Analysis (ICBDA), pp. 1-5.*

Paper II Yonas, Woldemariam. Predicting User Competence from Text. *Proceedings of The 21st World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI), pp. 147-152.*

Paper III Yonas, Woldemariam. Suna, Bensch. Henrik, Björklund. Predicting User Competence from Linguistic Data. *14th International Conference on Natural Language Processing (ICON-2017), pp. 476-484*

Paper IV Yonas, Woldemariam. Adam, Dahlgren. Designing a Speech Recognition-Named Entity Recognition Pipeline for Amharic within a Cross-Media Analysis Framework. *to be submitted.*

The following technical report is also produced, but not included in the thesis.

Paper V Henrik, Björklund. Johanna, Björklund. Adam, Dahlgren. Yonas, Woldemariam. Implementing a speech-to-text pipeline on the MICO platform. *Technical Report UMINF 16.07 Dept. Computing Sci., Umeå University,*
http://www8.cs.umu.se/research/uminf/index.cgi, 2016.

Financial support for this work is provided in part by the EU FP7 MICO project.

Acknowledgments

First of all, I exalt the most high God, the maker of heavens and earth, in the name of Lord Jesus Christ, for his mercy, grace, peace, strength, wisdom, knowledge, understanding, interventions and all spiritual blessings, and also for protecting me from the wickedness of the world during my studies.

I am deeply grateful for my supervisor Henrik Björklund for his excellent guidance and exceptional patience. It has been a great privilege to work with him and learn many professional qualities and ethics. I also would like to thank my co-supervisor Suna Bensch, for her wonderful support and constructive feedback during the development of this thesis as well as throughout my studies.

My special appreciation goes to Johanna Björklund, for providing me an opportunity to work in the MICO project. I am thankful for Frank Drewes for sharing his research experiences and constructive feedback during the research methodology course. I would like to thank Adam Dahlgren for his friendship and support during the work with the MICO project as well as Kaldi. I am thankful for all colleagues of the formal and natural language research group for the enjoyable Friday lunches, and social events. I would like to thank the whole community of the computing science department for the friendly working environment and unreserved technical support.

Furthermore, many thanks to my Ethiopian friends, Ewnetu and Selome for their kind support and helping me adapt and get to know student life in Umeå University.

Umeå, April 2018
Yonas Demeke Woldemariam

Contents

Chapter 1

Introduction

This thesis focuses on developing infrastructures for natural language analysis, intended to be integrated in an open-source cross-media analysis framework. This includes design and implementation of different Natural Language Processing (NLP) components, in particular in the areas of sentiment analysis and users competence analysis, and speech and named entity recognition.

NLP deals with the task of digitally processing and automating natural languages, occurring in the form of text and speech, and is a subfield of Computer Science and Artificial Intelligence, also closely related to Linguistics, Data Mining and Information Extraction fields. Some of the most used NLP systems are, for example, automatic grammar checker, automatic speech recognition, machine translation and so on. In the case of text analysis, NLP covers the whole spectrum of tasks from morphology analysis, stemming, part-of-speech tagging, to named entity recognition, sentiment analysis, topic modeling, automatic summarization and discourse analysis. Speech processing, spans from automatic speech recognition, speech dialog to speech generation from text.

While NLP attracted many researchers to contribute in the field since the 1950s, it presents a lot of challenges, potentially affecting the reliability of the NLP systems. The main challenging issues are variations across languages in general (syntax), ambiguities, domain and context.

The techniques developed for one language cannot be used for others due to various reasons, for example, capitalization is used as a very important clue to detect named entities for English, however, most Semitic languages such as Arabic and Amharic do not have that feature. Also due to other wide variations, it is hard to easily extend the effort used to build (computational) linguistic resources for well-studied languages such as English, Spanish and French, to under-resourced languages. This is one of the issues addressed in this thesis.

Ambiguity in NLP exists in different forms such as word-sense ambiguity (a word in a sentence might have more than one meaning), syntactic ambiguity (a sentence can be represented with multiple syntactic structures) and so on.

Ambiguity could potentially reverse the results returned by NLP systems, for instance, in sentiment analysis, a positive review could be misclassified as negative due to ambiguous words or phrases occurring in the review. Depending on the types of ambiguity, there are possible strategies, for example, morphological analysis to resolve lexical ambiguity. However, most of them use statistical models trained on large corpus, but lack sufficient contextual information for disambiguation.

NLP techniques and tools, in particular the supervised and data-driven ones, as they heavily depend heavily on specific domain-knowledge and thus their application is limited to closely related domains. For example, most sentiment analysis models are trained on movie reviews. As a result they perform poorly in forum discussion domains, which became evident from our experimental results [18].

Lastly, NLP applications are mostly designed to run in an environment where the input is usually an original (natural) text. However, within cross-media analysis solutions the input text is sometimes extracted from video content via a speech recognition component or from images via an optical character recognition component (OCR). In that case, unless the challenges are not sufficiently addressed, the text analysis components fail to process the extracted text due to the incompatibility of the format required by the text analysis components with the speech recognition or the OCR component. Thus, there is a demand for effective collaboration between the NLP components and other multimedia extractors in an orchestrated fashion. Thus, to meet the requirements posed by such collaborative environments new methods dealing with associated challenges need to be explored.

We discuss conceptual backgrounds on NLP in Chapter 2, NLP tasks in cross-media analysis frameworks in Chapter 3. The main contributions of our studies is summarized in Chapter 4 and, finally, the discussion of future directions in Chapter 5. We also attached the papers summarized in this thesis.

Chapter 2

Conceptual Backgrounds on NLP

Here, we describe core NLP tasks performed in general computational linguistic analysis and required for many applications as pre-processing or intermediate steps. We also briefly discuss *sentiment analysis* and *competence analysis*, which provides background knowledge for the areas that we explored and summarized in this thesis.

2.1 Text Analysis

An initial step in *natural language analysis or text mining* workflows, is to parse a document and put it into some kind of representations prior to actual text analysis tasks, and extract basic features, widely used and shared by most NLP applications. That potentially makes subsequent computations easier for extracting target information from textual data and determines its representation. We briefly describe such tasks that have been relevant for studies on *sentiment* and *competence analysis*.

Data cleaning involves removing noisy features such as XML tags, smileys and so on, from raw text and then generates a plain text. It might also include stop-words removal, and filtering other common words that are not relevant for, e.g. text classification or document retrieval, and lowercase transformation.

Tokenization splits an input document or text into a sequence of tokens. A token, for example, might be a word in word tokenization. There are several ways of doing that by using regular expressions containing non-alphanumeric characters. The resulting list of tokens often used by subsequent text analysis tasks such as stemming and part-of-speech tagging. For example, word tokenization segments the text "Models of natural language understanding by Bates" on whitespace and returns [‘Natural’, ‘Models’, ‘of’, ‘language’, ‘understanding’, ‘by’, ‘Bates’].

Stemming takes the word tokens returned during the tokenization phase and generates a morphological base form of the words by stripping the word suffixes. For example, the Porter stemming algorithm [12], which is considered as a de facto standard algorithm for English. The For the above tokenized text the stemming algorithm returns [‘Natural’, ‘Model’ , ‘of’, ‘language’, ‘understand’, ‘by’, ‘Bates’].

Part-of-speech tagging annotates each word in text with its syntactic category or part of speech(POS) such as noun, pronoun, verb, adverb and adjective. POS tagging algorithms (POS taggers) make use of linguistic rules along with dictionaries, or statistical models, to tag words with their POS tags. In case, words with multiple POS encountered, contextual information of the words can be used by POS taggers to disambiguate. For example, "influence" can be a noun in the phrase "the influence of postmodernism" or a verb in "moral reasoning is influenced by virtue".

Named entity recognition (NER) identifies entity mentions such as names of people, locations and organizations from text. For example, "Bates" is recognized as a person from the previous stemmed text.

Generating *n*-grams an n-gram is a sequence of tokens of length n. Ideally, capturing all possible sequences of tokens in a document may improve the performance of text classification and information retrieval systems, though it is computationally expensive.

Generating a document-term matrix is the task of representing a corpus of documents as a matrix where each document is represented with a row-vector containing the calculated frequency count of its tokens. The most widely used technique for constructing a document-term matrix is TF-IDF (term frequency–inverse document frequency).

Parsing is used to carry out syntactic analysis and extract information about the syntactic structure of text. For example, we use the Stanford probabilistic context-free grammar (PCFG) parser [7] for this purpose.

Extraction of number of tokens returns the frequency counts of tokens in each document and is a very important feature in probabilistic models, such as naive bayes [8].

Extraction of aggregate tokens length calculates the size of each document by aggregating the frequency counts of all tokens occurring in that document.

2.1.1 Sentiment Analysis

Sentiment analysis detects polarity and extracts expressed sentiments typically from opinion-oriented text such as comments in blog posts, movie reviews and product reviews. It allows to understand how people feel about, for example, the service provided by on-line companies, the headlines posted on news sites, political discussions going on social media, and so on. Thus, exploring methods to automatically analyze, extract, classify and summarize opinions from those texts would be enormously helpful to individuals, journalists, business and government intelligence and in decision-making. Some of the early research works in this area done by Pang et al. [11]. In their work different methods have been used for detecting the polarity of movie reviews. A survey on sentiment analysis algorithms and applications can be found in Medhat et al. [10], and state-of-the arts methods by Richard et al. [15].

In the task of sentiment analysis, the most prominent challenges include dealing with sarcasm and capturing the scope of negation in a statement. *Sarcastic statements* or ironic comments are hard to detect because they are too implicit and deep, strategically conveyed probably to affect audiences negatively. Regarding the scope of negation, unless properly determined, for example, using a negation-annotated corpus, a negation cue (such as ”never”, ”not”, and so on) could either negate only a single succeeding word or multiple words of a sentence, which results in variations on an overall sentiment of the sentence. While the problem of automatically identifying sarcastic sentences is studied by Dmitry et al. [5], using a semi-supervised classifier trained on datasets obtained from Twitter and Amazon, identifying the scope of negation investigated by Richard et al. [14] using the introduced neural networks-based method along with the Stanford Sentiment Treebank.

In literature, lexicon-based and machine learning-based, are the two broad approaches of sentiment analysis. Machine learning algorithms predict sentiment using learned models trained on opinion-annotated corpora. The lexicon-based approach determines the overall sentiment of a sentence by computing and aggregating the sentiment polarity of individual words in the sentence using dictionaries of words annotated with sentiment scores.

2.1.2 Competence Analysis

Basically, *competence analysis* attempts to discover the relationship between the text written by authors in connection with a specific task and their performance regarding that task. Unlike sentiment analysis, it is a less researched variant of text analysis. Competence analysis can take different forms, for instance, evaluating the quality of an essay [2], assessing the performance of medical students from their clinical portfolio [3] and so on. In our studies [17, 19], we explored assessing the proficiency of users in classifying images of different types of objects hosted on crowd source platforms.

There are a number of studies [9, 2, 4] related to competence analysis.

A comprehensive survey on existing state-of-the-art approaches for automatic essay scoring can be found in [2]. Regardless of the form of *competence*, most of these research works generally make use of NLP methods for analyzing authors text and extract linguistic features, and ML techniques for developing statistical models based on the linguistic features. These features include lexical (e.g. number of words), syntactic (e.g. frequency count of syntactic categories), and fluency features.

2.2 Machine-Learning Methods in NLP

We give a formal and brief description for the three ML methods used in our studies, naive bayes [8], decision trees [1] and K-nearest neighbor [20].

2.2.1 Naive Bayes

Naive Bayes (NB) is a probabilistic classifier and applied to several text classification problems [2]. Once trained with a corpus of documents, the NB model returns the most probable class for the input text based on Bayes' rule of conditional probability. First, the text (a document) needs to be defined and represented with a set of features. We assume that T is a set of training samples. Then NB takes a feature vector $\overrightarrow{d} = (f_1, \ldots, f_n)$ of the document. In the bag-of-words model each feature f_i for i=1...n represents the frequency count of each word/token. NB applies the following equation to predict the most likely class:

$$\underset{C}{\operatorname{argmax}} P(C|\overrightarrow{d}) \tag{2.1}$$

$$P(C|\overrightarrow{d}) = \frac{P(f_1, \ldots, f_n|C)P(C)}{P(f_1, \ldots, f_n)}. \tag{2.2}$$

The term $P(C|\overrightarrow{d})$ is the probability of $\overrightarrow{d}$ being in class C, defined as:

$$P(C|\overrightarrow{d}) \sim \frac{P(C)\prod_{i=1}^{n} P(f_i/C)}{P(f_1, \ldots, f_n)}. \tag{2.3}$$

Here the term $P(C)$ is the prior probability of class C and (f_i/C) is the conditional probability of f_i given class C. Since $P(f_1, \ldots, f_n)$ is the same for all classes. Then, the above equation can be reduced to:

$$P(C|\overrightarrow{d}) = P(C)\prod_{i=1}^{n} P(f_i/C) \tag{2.4}$$

The probability P over T is estimated based on word/token and class counting as follows:

$$P(C) = \frac{count(C)}{|T|}. \tag{2.5}$$

$$P(f_i/C) = \frac{count(f_i, C)}{TC}. \tag{2.6}$$

Here $count(C)$ returns the number of times that class C is seen in T, and $|T|$ is the total number of samples in the training corpus, TC is the total number of (words or tokens) in class C, $count(f_i, C)$ returns the number of times the word/token f_i seen in class C. For instance, in our study, to avoid *zero probabilities*, *Laplace correction (add-one smoothing)* has been used. That is a commonly used parameter smoothing technique which adds one to each count.

2.2.2 Decision trees

Decision trees (DT) is extensively used in a wide range of NLP applications for building tree structured predictive models for solving classification and regression problems. For the classification problems, the classes correspond to predefined categories have discrete values, for instance, in document categorization, the documents might belong to one of the following classes based on their subjects: "Computer Science", "Mathematics" and "Statistics". Whereas, the classes in the regression problems take continuous values, for example, in segmental duration prediction for text-to-speech systems, speech units of variable length can be assigned real values of duration based on their acoustic features. Decision trees built by a DT algorithm consist of the root node, which represents the most discriminatory feature in the training feature set, edges represent answers to the questions asked by internal nodes, and leaf nodes correspond to decisions [1]. To split training samples (T) with N number of classes and n number of features of the form, $(f_1, \ldots, f_n, C)$ into subtrees, the DT algorithm computes Entropy (H), which is the measure of homogeneity of T, and Information Gain (IG), which is the measure of a decrease in H.

Here are the equations for H and IG respectively:

$$H(T) = -\sum_{i=1}^{N} P(C_j) \log_2 P(C_j), \tag{2.7}$$

where N is the number of classes and the term $P(C)$ is the probability of class C_j. The IG for any f_i in a feature set characterizing T, defined as:

$$IG(T, f_i) = H(T) - \sum_{x \in X} P(x) \sum_{i=1}^{n} P(C_j|x) \log_2 P(C_j|x), \tag{2.8}$$

where X is a set of values of feature f_i in T, and the term $P(x)$ is the probability of $x \in X$.

During the construction of a decision tree, the feature yielding the highest IG taken by the DT algorithm to split the samples recursively until it reaches the stopping criteria set to limit the number of samples. The decision tree can be optimized using different techniques such as pruning, and also by varying model parameters such as maximum tree-depth and minimal gain.

The accuracy of decision trees can be further improved by utilizing ensemble methods, which result in a *boosted model.* For example, a gradient boosted model can be built by combining a series of *weak models* learned iteratively from the same training samples. At each iteration, the *the gradient boosted algorithm* tries to reduce the prediction error e.g. the root mean square error (RMSE) (the difference between predicted and actual values) of the previous model in the case of the regression problem, by optimizing the loss function that calculates RMSE using a development set.

2.2.3 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a non-parametric classifier. In a KNN algorithm, K represents the number of nearest neighborhood samples. Those samples belong to the class predicted by the algorithm. The nearest neighbors to input samples are obtained by using, for example, *Euclidean* distance. KNN has been used in many applications such as search engines [16], and pattern matching [20].

The Euclidean distance between the two feature vectors, $({f_1}^1, \ldots, {f_n}^1)$ and $({f_1}^2, \ldots, {f_n}^2)$ representing two documents $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ respectively is:

$$D(\overrightarrow{d_1}, \overrightarrow{d_2}) = \sqrt{\sum\nolimits_{i=1}^{n} ({f_i}^1 - {f_i}^2)^2}. \tag{2.9}$$

During the prediction phase, given $\overrightarrow{d}$, we find the k nearest neighbors to $\overrightarrow{d}$ in the training data. We assign $\overrightarrow{d}$ the class that is most common among these k example.

Chapter 3

NLP Tasks in Cross-Media Analysis Frameworks

To empower web search engines with concept-driven search facilities, they need to be supported with dynamic *cross-media analysis technologies.* Basically, *cross-media analysis frameworks* provide media analysis, metadata extraction and annotation services. Such frameworks potentially improve the searchability of media assets by semantically enrich them with the extracted metadata representing the hidden meanings. To support the analysis of various types of media such as text, image, audio and video, several extractors corresponding to these types need to be integrated and orchestrated in cross-media analysis frameworks. To support complex use-cases within cross-media platforms among other analysis components, mostly high attention is given for text-transcription and text-annotation tools such as *automated speech recognition (ASR)* and *named entity recognition (NER)* respectively, as the whole point is to make multimedia data as searchable as textual contents.

Although the NLP tasks discussed in Chapter 3 are important for processing texual content, the tools performing those tasks are implemented to effectively operate in NLP environments. As a result, introducing them in cross-media frameworks require a lot of efforts to design and develop various techniques, for instance, for enabling them to be able to use the data model shared within the frameworks for representing analysis results and effectively interact with other audio-visual analysis extractors and metadata storage and retrieval components.

In this chapter, we describe the MICO[1](Media in Context) platform as an example cross-media solution and the key NLP tasks within the platform.

[1] https://www.mico-project.eu

3.1 The MICO platform

MICO basically provides media analysis, metadata publishing, search and recommendation services. Its design is based on service oriented architectures (see Figure 3.1) where analysis components communicate and collaborate with each other in an automatic fashion via a service orchestration component (aka broker) to put a media resource in context. Its implementation is heavily based on open-source libraries, for example, semantic web technologies such as Apache Marmotta[2] and SPARQL-MM[3] have been used for storing the metadata annotation of analysis results in RDF format and querying the metadata respectively. The Apache Hadoop[4] distributed file system is used for binary data, and Apache Solr[5] for the full-text search.

MICO extractors can be divided into three groups with respect to the media type they analyze, namely audio, visual and textual extractors: We describe them briefly:

Visual extractors perform image analysis for detecting e.g., human faces and animals in images. Their models, particularly the animal detection extractors, are trained on the dataset obtained from the Zooniverse Snapshot Serengeti project [6].

Audio extractors include different speech analysis tasks such as detecting whether audio signals contain music or speech, and extracting audio tracks from video content and producing a transcription (we elaborate on this in the next section)

Textual extractors provide linguistic analysis services, including parsing, sentiment analysis, text classification and competence analysis and so on.

3.2 Automatic Speech Recognition (ASR)

Speech recognition is one of the most important NLP tasks for the analysis of spoken language in cross-media analysis solutions. It extracts a transcription (text) from an input audio a video recording. This allows the indexing and retrieval of spoken documents with a simple keywords search. However, to support advanced use cases, for example, searching video shots containing a person making a speech on a specific topic, the resulting transcription needs to be further analyzed with textual extractors. It also needs to be supported

[2] http://marmotta.apache.org

[3] http://marmotta.apache.org/kiwi/sparql-mm.html

[4] http://hadoop.apache.org

[5] http://lucene.apache.org/solr/

[6] https://www.snapshotserengeti.org

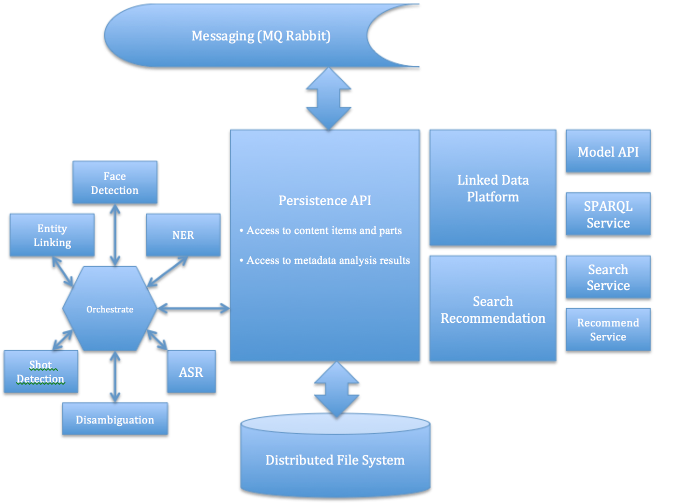


Figure 3.1: The Architecture of the MICO Platform, adopted from [13]

with other auxiliary components for extracting audio tracks and making the transcription accessible to the text analysis components.

Though speech recognition is an extensively studied problem and different techniques and tools are available, they fail to meet some of the requirements such as multi-lingual support and smooth interaction with other extractors of cross-media frameworks. For example, the ASR technology aimed to be used in MICO required support for English, Italian and Arabic. Unfortunately, it was only possible for English due to the lack of open-source language models. Compared to others language specific components, training the ASR model is quite costly due the requirement of a sufficiently large parallel corpus (speech and text). This problem is more apparent when it comes to computationally under-resourced languages. This is one of the problem explored in our thesis.

In practice, the entire speech recognition work-flow can be implemented and integrated into cross-media frameworks in various ways, obviously yielding different results in performance and transcription quality. There are also quite shared trends employed to manage the underlying interaction problem between multi-modal extractors. Within MICO, the ASR is implemented as a *speech-to-text pipeline*. The pipeline includes audio-demultiplexing, for extracting and down-sampling the audio signal from the video, speaker diarization for segmenting audio-tracks along with gender classification and speaker partitioning, speech transcription, for transcribing the audio signal into text. The resulting textual content outputted by the pipeline is further analyzed by text analysis components including the NER extractor.

3.3 Named Entity Recognition and Linking

In the context of cross-media analysis frameworks, the NER component plays the role of extracting and linking entity mentions, such as names of people, organization, places and so on, not only from textual content but also possibly from audio-visual content. For example, in the previous use case, NER extracts and associates the name of the person in the video to concrete real world entities using semantic knowledge bases such as DBpedia. The *Entity linking* involves disambiguating and tagging extracted items with the URI (Uniform Resource Identifier) reference of the corresponding objects in a knowlege base, which potentially enhances the semantic enrichment of the media being analyzed. For example, given the video where the term "Washington" is mentioned, which may refer different entities such as "Washington D.C "(place), and "George Washington"(person) and so on, then the entity linking service disambiguates the term using the associated contextual information. NER also serves as a subtask for other text analysis tasks such as sentiment analysis, text classification and document summarization.

The NER extractor works with audio-visual extractors such as OCR for extracting entities from subtitles and captions, to define complex workflows relevant for cross-media applications. It also closely works with the ASR component and forms an analysis chain called *ASR-NER pipeline* (shown in Figure 3.2) for extracting entities from spoken documents as well as videos, and annotating and indexing them with the extracted textual metadata. While applying the NER extractor on original (natural) textual content is fairly simple, named entity extraction on speech transcripts is a challenging task and prone to errors due to a lack of linguistic features in the transcripts such as punctuations and capitalization, which are very important clues for NER. For example, the

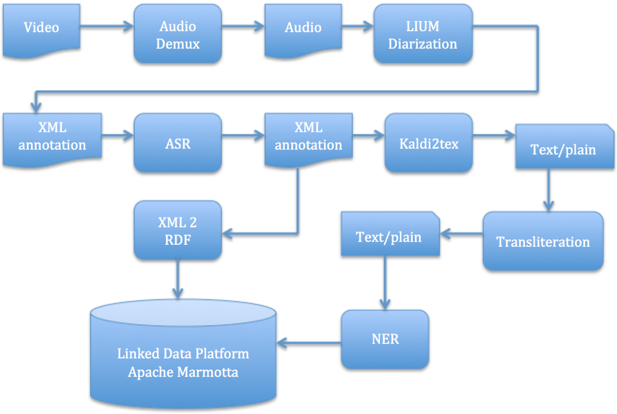


Figure 3.2: An ASR-NER Pipeline within a Cross-media Analysis Framework

authors in [6], introduced a method to normalize and recover the speech transcripts using a monolingual statistical machine translation system.

Chapter 4

Summary of Contributions

Our contributions cover three related sub-topics of NLP. The first one is on sentiment analysis in the context of a cross-media analysis framework. There, we deal with the problem of evaluating, implementing and integrating sentiment analysis methods. The second one focuses on assessing user performance in a specific task using different types of linguistic features extracted from a science crowd sourcing platform hosting many projects. We developed an algorithm that estimates the proficiency of users and annotates their text with computed competence. We describe the key contributions from the studies of sentiment and competence analysis in the following section. Lastly, we address the multi-lingual issue of cross-media analysis solutions. We explore developing computational linguistic infrastructures for one of the under-resourced languages i.e., Amharic.

4.1 Sentiment and Competence Analysis

4.1.1 Paper I: *Sentiment Analysis in a Cross-media Analysis Framework.*

In this paper, we investigate the problem of applying sentiment analysis methods on crowd-sourced discussion forum posts. The corpus (chat messages) for this study is obtained from Snapshot Serengeti, which is one of the projects hosted by the world's largest crowd sourcing platform Zooniverse. Researchers in Snapshot Serengeti aim to investigate classifying wildlife in Tanzania Serengeti National Park into species. In the Park, several cameras are installed to capture images of animals. Those images are posted on the on-line platform of Snapshot Serengeti to be classified by volunteers. Moreover, the platform also has a forum where the volunteers discuss their respective classifications.

Unlike other types of discussion forums where their posts often characterized by expressed sentiment, Snapshot Serengeti's texts contain mostly explanatory information about observed images. Thus, studying how sentiment analysis methods behave on such type of texts and empirically choose the best method, is of particular interest. Then we aim to implement and integrate the selected method into the MICO platform.

We compare two broad categories of sentiment analysis methods, namely lexicon-based and machine-learning approaches. From the implementation point of view, we need to find sentiment analysis tools that potentially fit with the working infrastructures of the underlying cross-media framework. For that reason, we run the built-in lexicon based algorithm of Apache Hadoop and the RNTN (Recursive Neural Tensor Network) based algorithm of Stanford Core NLP. We found that the ML model outperforms the lexicon-based by 9.88% accuracy on variable length positive, negative, and neutral comments. However, the lexicon-based shows better performance on classifying positive comments. We also obtained that the F1- score by the lexicon-based is greater by 0.16 from the ML.

4.1.2 Paper II: *Predicting User Competence from Text* Paper III: *Predicting User Competence using Linguistic Data*

In these two articles, we go beyond extracting user sentiment, done in Paper I, to extract user competence from forum discussion posts. The papers target the users of the two sub-projects of Zooniverse, namely Snapshot Serengeti and Galaxy Zoo. Paper III [19] is an extension of Paper II in terms of the linguistic features extracted from text and the methods used to analyze the data.

In Paper II [17], we explore the possibility of learning user performance in classifying images, from the associated text posted by the user. A weighted majority scheme was used as a ground truth to calculate the competence of the users. Then, each user is annotated with a competence value ranging from 0 to 1 along with the text aggregated from his/her posts to form a document. The bag-of-words model is used to represent the documents, also a bi-gram feature is extracted.

We evaluate and compare the performance (regarding accuracy and F-measure) of the three ML methods, Naive Bayes (NB), Decision Trees (DT) and K-nearest neighbors (KNN), trained on the same corpus, but in two different experimental settings: baseline and calibrated. In the former case, the users are divided into 5 levels of competence via partitioning the competence scale into 5 equal sizes: very incompetent, incompetent, average, competent, very competent based on their competence values, ranging [0.00, 0.2], (0.20, 0.40], (0.40, 0.60], (0.60, 0.80] and [0.80, 1.00] respectively. In the latter case, we attempted to calibrate the competence scale to have only three categories to reduce the class imbalance problem, which improved the accuracy of the models to some extent. The baseline results show, that regarding accuracy, DT outperforms NB and KNN by 16.00%, and 15.00% respectively. Regarding F-measure, K-NN outperforms NB and DT by 12.08% and 1.17%, respectively. It turns out that while adding the bi-gram feature dramatically improved the performance of the NB model, adding the number of classifications of a user improved the performance of the KNN and the DT models significantly.

In Paper III [19], we extended Paper II [17] with further analysis of the problem using new strategies and additionally extracted linguistic features from different but related domain data. We also divided the users based on their distributions over the competence scale, so that in all categories (levels) of competence, there are almost equivalent number of users, compared to the strategy used in Paper II [17], which completely solves the class-imbalance problem. The extracted linguistic features include *syntactic categories*, *bag-of-words*, and *punctuation marks*. Given the individual feature sets and their combinations turn out to give 6 different feature set configurations: Bag-of-Words (BoW), punctuation marks (Pun), punctuation marks with Bag-of-Words (Pun+BoW), syntactic, syntactic with Bag-of-Words (Syn+BoW), and the combination of BoW, punctuation mark and syntactic (BoW+Pun+Syn). We trained three classifiers using the resulting feature sets: k-nearest neighbors, decision trees (with gradient boosting) and naive Bayes. Before we evaluate the performance (regarding accuracy and F-measure) of the classifiers, a statistical significance test is run to make sure that the trained classifier models give results that are significantly better than chance. The evaluation of the models are carried out using both Galaxy Zoo and Serengeti Snapshot test sets, which ensures that the results can be generalized to other crowd-sourced projects. The overall results show that the performance of the classifiers varies across the feature set configurations.

4.2 Speech and Named Entity Recognition

4.2.1 Paper IV: *Designing a Speech Recognition-Named Entity Recognition Pipeline for Amharic within a Cross-Media Analysis Framework. Manuscript, to be submitted for publication*

One of the major challenges that are inherently associated with cross-media analysis frameworks, is addressing the multi-lingual issue. Within these frameworks, there are several language dependent analysis components such as textual and spoken data extractors, that require trained models of different natural languages. Here, we investigate adapting language specific components of the MICO platform, in particular, speech recognition and named entity recognition for Amharic, as other extractors depend and build on them.

We design an ASR-NER pipeline (analysis workflow) that includes three main components: ASR, transliterator and NER. To develop the ASR system, we explored and applied three different modeling techniques used for speech signal analysis, namely Gaussian Mixture Models (GMM), Deep Neural Networks (DNN) and the Subspace Gaussian Mixture Models (SGMM) using acoustic features such as Mel-frequency cepstrum coefficients (MFCCs) features, followed by linear discriminant analysis (LDA) and transformation, maximum likelihood transform (MLLT). The models have been evaluated with the same test set with 6203 words using the Word Error Rate (WER) metric, and obtained an accuracy of 50.88%, 38.72%, and 46.25% for GMM, DNN, SGMM respectively. For the NER component, we use the existing OpenNLP-based NER model developed for Amharic, though trained on very limited data. While the NER model was trained with the transliterated form of the Amharic text, the ASR is trained with the actual Amharic script. Thus, for interfacing between ASR and NER, we implemented a simple rule-based transliteration program that converts an Amharic script to its corresponding English transliteration form.

Chapter 5

Future Work

While working on the problems investigated and described in this thesis, we identified a number of potential gaps for future investigations, however, our immediate plans to extend the thesis include improving the Amharic ASR using a new language model and further studies of competence analysis using formal language models, particularly, cooperating distributed grammar systems.

We are also interested to work on the possible solutions suggested to tackle the challenges that are extensively addressed in Paper III [19]. These solutions are, utilizing semi-supervised bootstrapping methods and topic modeling techniques to approach the competence analysis problem. The former helps reduce the dependence on a majority-vote scheme and the latter enables to generate domain-specific words, which in turn become part of the linguistic features. Also, to further enrich the syntactic features, we can apply dependency parsing to extract universal dependencies. The resulting methods can also be applied on question-answers frameworks to extract various types of information, for instance, the quality of questions posted by users.

Bibliography

[1] L. Breiman et al. *Classification. and Regression Trees.* Monterey, CA: Wadsworth Brooks/Cole Advanced Books Software, 1984.

[2] H. Chen and B. He. “Automated Essay Scoring by Maximizing Human-machine Agreement”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* 2013, pp. 1741–1752.

[3] Y. Chen et al. “Automated Assessment of Medical Students’ Clinical Exposures according to AAMC Geriatric Competencies”. In: *AMIA Annual Symposium Proceedings Archive.* 2014, pp. 375–384.

[4] M. Dascalu, E-V. Chioasca, and S.A. Trausan-Matu. “ASAP – An Advanced System for Assessing Chat Participants”. In: *AIMSA: International Conference on Artificial Intelligence: Methodology, Systems. and Applications.* Vol. 5253. Lecture Notes in Computer Science. Springer, 2008, pp. 58–68.

[5] D. Davidov, O. Tsur, and A. Rappoport. “Semi-Supervised Recognition of Sarcastic Sentences in Twitter. and Amazon”. In: *In Proceedings of the 14th Conference on Computational Natural Language Learning.* 2010, pp. 107–116.

[6] J. Grivolla et al. “The EUMSSI project – Event Understanding through Multimodal Social Stream Interpretation”. In: *Proceedings of the 1st International Workshop on Multimodal Media Data Analytics co-located with the 22nd European Conference on Artificial Intelligence (ECAI 2016).* 2016, pp. 8–12.

[7] D. Klein and C.D. Manning. “Accurate Unlexicalized Parsing”. In: *Proceedings of the 41st Meeting of the Association for Computational Linguistics.* 2003, pp. 423–430.

[8] D. Lewis. “Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval.” In: *Proceedings of the European Conference on Machine Learning (ECML).* 1998, pp. 4–15.

[9] D.S. McNamara, S.A. Crossley, and P.M. McCarthy. “Linguistic Features of Writing Quality”. In: *Written Communication* 27.1 (2010), pp. 57–86.

[10] W. Medhat, A. Hassan, and H. Korashy. “Sentiment Analysis Algorithms. and Applications: a Survey”. In: *Ain Shams Eng J* 5.4 (2014), pp. 1093–1113.

[11] B. Pang, L. Lee, and S. Vaithyanathan. “Thumbs up?: Sentiment Classification using Machine Learning Techniques”. In: *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language processing*. 2002, pp. 79–86.

[12] M.F. Porter. “An Algorithm for Suffix Stripping”. In: *Program* 14.3 (1980), pp. 130–127.

[13] S. Schaffert and S. Fernandez. *D6.1.1 MICO System Architecture. and Development Guide*. Deliverable, MICO. 2014.

[14] A. Socher et al. “Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank”. In: *in In ACL Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 354–368.

[15] R. Socher et al. “Semantic Compositionality Through Recursive Matrix-Vector Spaces”. In: *In Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2012.

[16] J. Suchal and P. Návrat. *Full Text Search Engine as Scalable k-Nearest Neighbor Recommendation System*. Vol. 331. In: Bramer M. (eds) Artificial Intelligence in Theory and Practice III. IFIP AI 2010. IFIP Advances in Information and Communication Technology. Berlin Heidelberg: Springer, 2010.

[17] Y. Woldemariam. “Predicting Competence from Text”. In: *Proceedings of The 21st World Multi-Conference on Systemics, Cybernetics. and Informatics (WMSCI)*. 2017, pp. 147–152.

[18] Y. Woldemariam. “Sentiment Analysis in a Cross-Media Analysis Framework”. In: *2016 IEEE International Conference on Big Data Analysis (ICBDA)*. 2016, pp. 1–5.

[19] Y. Woldemariam, S. Bensch, and H. Björklund. “Predicting User Competence from Linguistic Data.” In: *14th International Conference on Natural Language Processing (ICON-2017)*. 2017, pp. 476–484.

[20] Y. Wu, K. Ianakiev, and V. Govindaraju. “Improved K-Nearest Neighbor Classification”. In: *Pattern Recognition* 35.10 (2002), pp. 2311–2318.

I

Sentiment Analysis in A Cross-Media Analysis Framework

Yonas Woldemariam
Department of Computing
Science Umea University
Umea, Sweden
e-mail: yonasd@cs.umu.se

Abstract—**This paper introduces the implementation and integration of a sentiment analysis pipeline into the ongoing open source cross-media analysis framework. The pipeline includes the following components; chat room cleaner, NLP and sentiment analyzer. Before the integration, we also compare two broad categories of sentiment analysis methods, namely lexicon-based and machine learning approaches. We mainly focus on finding out which method is appropriate to detect sentiments from forum discussion posts. In order to conduct our experiments, we use the apache-hadoop framework with its lexicon-based sentiment prediction algorithm and Stanford coreNLP library with the Recursive Neural Tensor Network (RNTN) model. The lexicon- based uses sentiment dictionary containing words annotated with sentiment labels and other basic lexical features, and the later one is trained on Sentiment Treebank with 215,154 phrases, labeled using Amazon Turk. Our overall performance evaluation shows that RNTN outperforms the lexicon-based by 9.88% accuracy on variable length positive, negative, and neutral comments. How- ever, the lexicon-based shows better performance on classifying positive comments. We also found out that the F1-score values of the Lexicon-based is greater by 0.16 from the RNTN.**

Keywords-sentiment analysis; cross-media; machine learning algorithm; lexicon-based; neural network; (key words)

I. INTRODUCTION

A massive volume of both structured and unstructured mul- timedia data is being uploaded on the Internet due to rapidly growing ubiquitous web access over the world. However, analyzing those raw media resources to discover their hidden semantics is becoming a challenging task. As a result, it is difficult to retrieve the right type of media to satisfy multimedia content consumers. So, improving the searchability of multimedia contents on the web is one of the most appealing demands, especially for online audio/video content providers. Even if there are a lot of effective approaches for indexing textual contents, they cannot be applied to index media type such as audio and video, unless we transform them to some form of text, and add advanced metadata annotations using contextual information around the target media. This problem motivated for the genesis of the ongoing EU research project called Media in Context (MICO). MICO mainly aims at providing cross-media analysis framework, including orchestrated chain analysis components to extract semantics from the media in a cross-media context (eg. a web page containing text, image, audio, video, metadata and so on).

We are mainly concerned with the textual analysis aspect of MICO, including sentiment and discourse analysis, language identification, and named entity recognition. Sentiment analysis copes with the task of opinion mining from text. With the growth of user generated texts on the web, exploring the method to automatically extract and classify opinions from those texts would be enormously helpful to individuals, business and government intelligence and in decision-making. Some of the early research works in this area include [1], [2], in these works different methods have been used for detecting the polarity of product reviews and movie reviews respectively.

In general, sentiment analysis methods are classified into lexicon-based [3] and machine learning-based [4], [5]. Machine learning methods make use of learning algorithm and classifier models trained on a known dataset. The lexicon-based approach involves calculating sentiment polarity using dictionaries of words annotated with sentiment scores.

The general goal of this study is to assess the available sentiment analysis technologies and adapt to MICO. In order to achieve the goal, we compare these two broad categories of sentiment analysis methods regarding to their prediction accuracy and find out which method outperforms the other. We chose our test case to be Zooniverse (https://www.zooniverse.org,) forum discussion domain. Zooniverse is an online plat- form where volunteers contribute for scientific discovery for its several projects. One of its projects is Snapshots Serngeti (http://www.snapshotserengeti.org,) the purpose is to study animals in Tanzania Serengeti National Park, volunteers go to their website to analyze and classify animals into species, discuss about their classification and generally about the images, on the forum posts. Our focus is to run sentiment analysis on texts extracted from the forum to help them assess what the volunteers feel about the quality of the images and generally about their services. Unlike the comments found in social media such as Twitter, the nature of the texts we get from Serengeti Snapshot is highly characterized by descriptions about observed images rather than explicit opinions. So studying sentiment analysis with such kind of text creates its own new research challenges due to its unique features and worth to observe how the sentiment analysis methods behave on these dataset. In order to conduct our experiments, lexicon- based sentiment prediction algorithm and Recursive Neural Tensor Network (RNTN) [5] model are chosen. The former is implemented on single node version of Hadoop platform, the dictionary contains sentiment words annotated with sentiment scores, and the later one is trained on Sentiment Treebank containing 215,154 phrases, labeled using Amazon Turk. We found that RNTN outperforms lexicon- based by 9.88% accu- racy. In order to give the whole picture of the comparison, we have calculated other measures such as precision, recall and F1-score.

The remainder of this paper is organized as follows: Section II presents related literature review. Section III gives an overview of sentiment analysis component within the MICO architecture. Section IV and V discuss two selected methods to be compared. Section VI discusses evaluation and results. Finally, the last section briefly indicates the directions for future research.

II. Related Litrature Review

Even though there are several research works [2], [4], [6] which compare methods for sentiment analysis, most of them focus on comparing different machine learning methods. There are a few comparative studies [7], [8] on lexicon-based versus machine learning approaches. In [7], twitter testing dataset with a total of 1,000 tweets used to undertake comparison be- tween lexicon-based and machine learning approaches. After data pre-processing steps such as data cleaning, stemming, part of speech (POS) tagging, and tokenization, they run tests using Support Vector Machine (SVM), Maximum Entropy (ME), Multinomial Naive Bayes (MNB), and k-Nearest Neighbor (k- NN) machine learning techniques. Sentiwordnet has been used for lexicon-based sentiment classification. The result shows that machine learning methods produce better accuracy rate than lexical based approach. As they stated, the significant influence from lexical database has been set as reference in determining positive and negative opinion that means the lexical based method highly depends on the occurrence of the sentiment words present on the database. Another comparison study is conducted in [8], using 1,675 sentences from political news domain, the dataset is divided into, 1,137 positive and 538 negative sentences. After data cleaning, the authors ap- plied tokenization, stop word removal, lemmatization and POS tagging using natural language tool kit (NLTK) and Stanford POS tagger The lexical based was implemented using Senti- WordNet and Naive Bayes (NB) and Support Vector Machine (SVM) machine learning algorithms were implemented using WEKA. Among the methods the best F-measure shown by SVM. Our study aims at presenting a general comparison of two sentiment analysis methods (lexicon-based and supervised structured machine learning technique). The experiment is carried out by implementing sandboxed version of apache- hadoop and Stanford coreNLP library on sample Zooniverse dataset. As hadoop and Stanford coreNLP are being used in the cross-media software project, which motivated us to focus on the two methods.

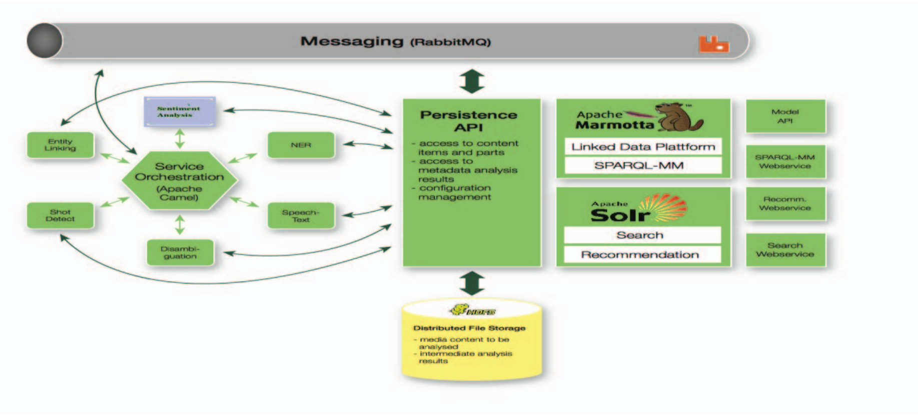


Figure 1. MICO General Architecture, adopted from [9].

III. An Overview of the Sentiment Analisis Component in A Mico Framework Architecture

The MICO framework uses a distributed service-oriented architecture (illustrated in Figure 1.), analysis components run independently and share communication and persistence infrastructure. Basically, the main services provided by the framework include, media analysis, search and recommendation. Once analysis components get registered with the framework and up running, the user can load a content item with its context. The service orchestration component notifies the respective analysis components about the input using its execution plan build as a result of service registration. The intermediate analysis results are stored with the metadata of the input in the persistence component, to enrich the existing basic metadata. Up on finishing processing the input content item, the final result is made available for further processing [9].

The input for a sentiment analysis component is a set of documents (or just a text from speech to text component within the framework), such as a HTML documents, news or movie reviews, comments from blog posts, or a text document in any format. The input is cleaned and pre-processed by chat room cleaner module, which removes non-standard characters and repeated spaces, and produces a plain text. Then the sentiment analysis uses its natural language processing sub component for tokenization, stemming, split into sentences, and so forth. Then the output texts are sent to the sentiment computation module which annotates them using the dictionary or machine-learning approaches, which includes annotations with sentiment polarity (positive/negative) of each word. The output of the sentiment analysis component is the annotations which can be attached to whole document.

IV. Lexicon-based Sentiment Classification Using Hadoop

Apache Hadoop (https://hadoop.apache.org), serves as big data solution for the processing of unstructured and complex sets of data. It uses the divide and rule methodology for processing through its parallel programming models. It mainly provides the Hadoop Distributed File System (HDFS) store the processed data. Apart from HDFS, Hadoop has several components and services including lexicon-based sentiment analysis. The main advantages we gain from this technology are big data analysis support and sentiment analysis service without having to prepare our own dictionary.

A. Data Pre-processing

Before we run lexicon-based algorithm for sentiment com- putation, we carried out the following pre-processing tasks:

1) Load the Snapshot Serengeti posts in CSV format into the HDFS.

2) Convert the raw posts into a tabular format.

3) Transform the data into a format that can be used for analysis.

B. Lexicon-based Algorithm

These are the major steps in the Algorithm 1.

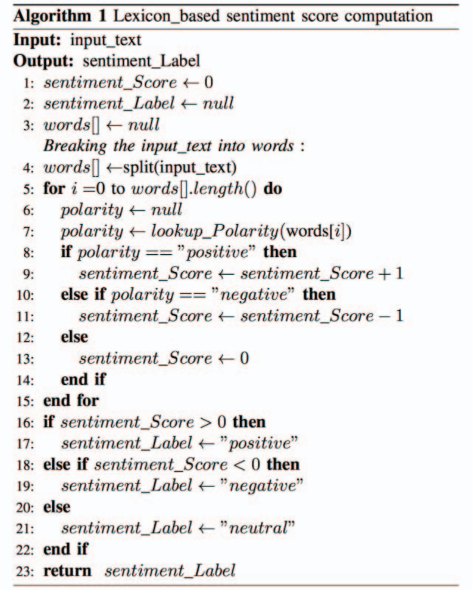
1) Tokenize the sentences into individual words.

2) Assign the polarity (positive, negative or neutral) for each word by using the sentiment dictionary.

3) Calculate the sum polarity value of all words within a sentence(s)

4) Compare the result with 0 and if result is greater than 0, then the sentiment is ‘positive ’or if result is equal to 0, then the sentiment is ‘negative ’, otherwise it is ‘neutral ’

5) Assign the sentiment value (2 for positive, 1 for neutral and 0 for negative) for the whole sentence



Algorithm 1 Lexicon_based sentiment score computation

```
Input: input_text
Output: sentiment_Label
sentiment_Score ← 0
sentiment_Label ← null
words[] ← null
Breaking the input_text into words :
words[] ←split(input_text)
for i =0 to words[].length() do
    polarity ← null
    polarity ← lookup_Polarity(words[i])
    if polarity == "positive" then
        sentiment_Score ← sentiment_Score + 1
    else if polarity == "negative" then
        sentiment_Score ← sentiment_Score − 1
    else
        sentiment_Score ← 0
    end if
end for
if sentiment_Score > 0 then
    sentiment_Label ← "positive"
else if sentiment_Score < 0 then
    sentiment_Label ← "negative"
else
    sentiment_Label ← "neutral"
end if
return sentiment_Label
```

V. Stanford Sentiment Treebank

In [5], Stanford Sentiment Treebank and Recursive Neural Tensor Network (RNTN) are introduced. The Treebank contains fully labeled parse trees constructed from the corpus of movie reviews that allows for a complete analysis of the compositional effects of sentiment in language. The main reason we use RNTN as machine-learning technique is, it has been already trained so we do not need to have labeled dataset for training purpose.

For the case of RNTN, we use already trained sentiment model, so we only need to extract texts from the Snapshot Serngeti database dump without being too much engaged with the text preprocessing tasks. Here is the description of Algorithm 2:

1) Tokenize sentences into individual words which are represented as a numeric vector

2) Lemmatize each word into their basic forms

3) Tag words with part of speech tagger (POS)

4) Parse sentences into their constituent subphrases and build a syntactic tree

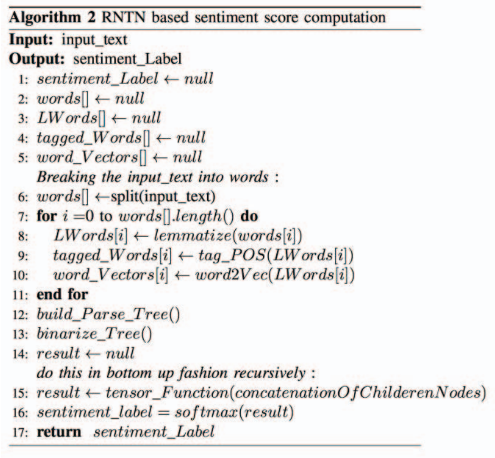
5) Binarize the tree, so that any parent node will have a maximum of 2 child nodes

6) Classify the sentences sentiment in a bottom up fashion using tensor-based composition function. The compo- sitionality function concatenates the vector of the two child nodes for each parent node, transforms the

vector resulted from the concatenation and analyse similarity.

7) The resulting vector is given to the softmax () classifier which computes its label probabilities, then the maxi- mum probability value will be returned as the sentiment label of the tree (sentence).

A. RNTN Algorithm



Algorithm 2 RNTN based sentiment score computation

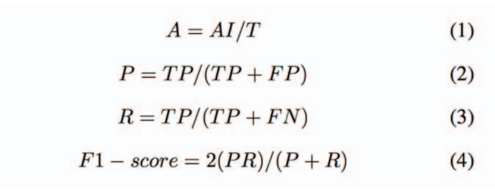
Input: input_text

Output: sentiment_Label

```
$sentiment\_Label \leftarrow null$
$words[] \leftarrow null$
$LWords[] \leftarrow null$
$tagged\_Words[] \leftarrow null$
$word\_Vectors[] \leftarrow null$
Breaking the input_text into words :
$words[] \leftarrow$split(input_text)
for $i$ =0 to $words[].length()$ do
    $LWords[i] \leftarrow lemmatize(words[i])$
    $tagged\_Words[i] \leftarrow tag\_POS(LWords[i])$
    $word\_Vectors[i] \leftarrow word2Vec(LWords[i])$
end for
$build\_Parse\_Tree()$
$binarize\_Tree()$
$result \leftarrow null$
do this in bottom up fashion recursively :
$result \leftarrow tensor\_Function(concatenationOfChilderenNodes)$
$sentiment\_label = softmax(result)$
return $sentiment\_Label$
```

VI. Experimental Evaluation and Discussion

In order to evaluate the performance of the two methods (Lexicon-based and RNTN), we randomly chose 600 sample tweets from Zooniverse, particulary from Serengeti Snapshot forum posts. The dataset has been made to contain 200 positive, 200 negative and 200 neutral tweets from each class, which are annotated by human judge. We apply commonly used performance metrics [10] in sentiment analysis. These are Accuracy (A), Precision (P), Recall (R) and F1-score. Precision measures the exactness of a classifier. A higher precision means less false positives (FP) (explained with equa- tion (1)), while a lower precision means more false positives. Recall measures the completeness, or sensitivity, of a classifier. Higher recall means less false negatives (FN) (explained with equation (2)), while lower recall means more false negatives. F1-score is harmonic mean of precision and recall, 1 its an ideal value, where as 0 is its minimum value.



$$A = AI/T \tag{1}$$

$$P = TP/(TP + FP) \tag{2}$$

$$R = TP/(TP + FN) \tag{3}$$

$$F1 - score = 2(PR)/(P + R) \tag{4}$$

Where AI is the number of accurately predicted instances, T is the total number of instances,TP is the number of accurately predicted positive instances, FP is the number of incorrectly predicted as positive instances and FN is the number of posi- tive instances, but incorrectly predicted as negative instances.

TABLE I. Evaluation Results of the Lexicon-based and RNTN

Metrics	Lexicon-based	RNTN
Accuracy	38.45	48.34
Precision	0.63	0.82
Recall	0.96	0.46
F-score	0.74	0.59

As experimental evaluation shown in table I, the RNTN method outperforms lexicon-based by 9.88%, it is just the overall accuracy. However, the lexicon-based shows better performance on positive comments. The Lexicon-based also

Scores nearly a perfect R, that means every positive instance (which does not include reversed negative instance “not bad”) is correctly classified. Even if a wide gap has been shown by the two methods in terms of P and R, they have quite closer F1-score value, which makes sense as R does not show a measure of false negative.

We also observed that stronger sentiment often builds up in longer phrases and the majority of the shorter phrases are neutral, which supports with the claim, demonstrated in [5]. It has been hard to classify short comments, for example some comments have just only hash tags with a single word. Mostly, these comments tend to be classified as neutral. Some of the comments are really hard to be classified even by human due to their ambiguity. We have to be careful what aspects and context to consider, for example the comment might explain the scene on the image very well, that means the volunteer has got reasonably clear image to discuss, so from the quality point of view, we classify the comment as positive, on the contrary, the comment does not bear any explicit opinion thus, which leads us to classify it to be neutral. That is one of the potential challenges of this study.

Another interesting fact is, unlike to lexicon-based algorithm, RNTN has a potential to capture negation and learn the sentiment of phrases following the contrastive conjunction “but”. In the case of lexicon-based, the major reason for the prediction errors is the algorithm fails to understand the context of the words including negation. In general, the performance lexicon-based algorithm could be improved by capturing the context of the words and stemming the input words into their basic form. In the case of RNTN, the main source of the prediction errors is the mismatching of domain knowledge between training dataset and test dataset. The training dataset is collected from movie reviews where as the test dataset is obtained from citizen-science domain; as a result the algorithm is challenged to recognise some unseen positive/negative phrases specific to the domain. Therefore, the straightforward approach to improve the prediction accuracy is to further train the RNTN model on Snapshot Serngeti posts.

VII. FUTURE WORK

For this study, we just focused on the comparison of two sample methods from each broad category of sentiment analysis approaches with limited test dataset. In the future, we are planning to experiment with other kind of methods such as, Naive Bayes, and Support Vector Machines. We are also interested to go beyond positive/negative polarity detection, and extend our work to extract other emotional knowledge from text such as the confidence and competence of the Snapshot Sernget users while they discuss in the forms.

ACKNOWLEDGMENTS

We acknowledge the financial support from the EU FP7 MICO project. We also thank the Zooniverse people for providing the dataset used in this study. Finally, we thank Adil (student of Umea□ University), for labeling the test dataset.

REFERENCES

[1] B.Pang and L.Lee, “Opinion mining and sentiment analysis,” vol. 3, no. 1-2, 2008.

[2] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up sentiment classification using machine learning techniquest,” in In ACL Conference on Empirical Methods in Natural Language Processing, 2010, pp. 354-368.

[3] C. Musto, G. Semeraro, and M. Polignano, “A comparison of lexiconbased approaches for sentiment analysis of microblog posts,” on 8th International Workshop on Information Filtering and Retrieval, 2014.

[4] G.Vinodhini and R.Chandrasekaran, “Sentiment analysis and opinion mining: A survey,” International Journal of Advanced Research in Computer Science and Software Engineering., vol. 2, no. 2277 128X, 2012.

[5] R.Socher, A.Perelygin, J. Wu, J.Chuang, C.Manning, A. Ng, and C.Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in Empirical Methods in Natural Language Processing, 2013.

[6] P. Gonalves, M. Arajo, F. Benevenuto, and M. Cha, “Comparing and combining sentiment analysis methods,” in In Proceedings of COSN, 2013, pp. 27-38.

[7] W. Maharani, “Microblogging sentiment analysis with lexical based and machine learning approaches,” in Information and Communication Technology (ICoICT), 2013 International Conference of. IEEE, 2010, pp. 439-443.

[8] S.Padmaja, S.Sameen, and S.Bandu, “Evaluating sentiment analysis methods and identifying scope of negation in newspaper articles,” vol.3 , No.11, no. 11, 2014.

[9] S. Schaffert and S.Fernandez, “D6.1.1 mico system architecture and development guide.” MICO, 2014.

[10] Olson, D. L, and Delen, Advanced data mining techniques. Dursun:Springer, 2008.

II

Predicting User Competence from Text

Yonas WOLDEMARIAM

Department of Computing Science
Umeå University, SE-901 87 Umeå, Sweden
yonasd@cs.umu.se

ABSTRACT

We explore the possibility of learning user competence from a text by using natural language processing and machine learning (ML) methods. In our context, *competence* is defined as the ability to identify the wildlife appearing in images and classifying into species correctly. We evaluate and compare the performance (regarding accuracy and F-measure) of the three ML methods, Naive Bayes (NB), Decision Trees (DT) and K-nearest neighbors (KNN), applied to the text corpus obtained from the Snapshot Senrengeti discussion forum posts. The baseline results show, that regarding accuracy, DT outperforms NB and KNN by 16.00%, and 15.00% respectively. Regarding F-measure, K-NN outperforms NB and DT by 12.08% and 1.17%, respectively. We also propose a hybrid model that combines the three models (DT, NB and KNN). We improve the baseline results with the calibration technique and additional features. Adding a bi-gram feature has shown a dramatic increase(from 48.38% to 64.40%) of accuracy for NB model. We achieved to push the accuracy limit in the baseline models from 93.39% to 94.09%.

Keywords:text analysis, NLP, machine-learning, naive bayes, decision trees, and K-nearest neighbors

1 Introduction

We evaluate and compare machine learning (ML) based models that predict the competence level of users from the text they have written. The main purose of this study is to identify an effective model and integrate into a cross-media analysis framework as a meta data extractor. In general the model can be used in other platforms where the proficiency evaluation of users is needed from their posted texts.

We consider texts obtained from Snapshot Serengeti[1] (SNS) discussion forum posts. The selected ML models are Decision Trees (DT), Naive Bayes (NB), and K-nearest neighbors (KNN). They have been proposed to be studied in our earlier work [13], we studied two sentiment analysis methods, namely the lexicon-based and recursive neural tensor network by applying them on the SNS forum posts. As part of the extension of [13], we study the problem of competence analysis as an advanced form of simple sentiment (positive/negative)detection and analysis by using the same dataset. The methods resulted from these studies are intended to support text analysis tasks in MICO[2]. MICO is an emerging soloution for analyzing, annotating and publishing media resources.

We analyze the texts generated by volunteer users of the SNS. Users are provided with randomly selected images of the wildlife in the Serengeti National Park, Tanzania, and expected to classify each image into one of 48 species [5]. Users then discuss what they observe in each image with their own texts in the forum. Administrators of the SNS project are interested to assess how well their users perform in classifying images, to understand and manage the users better. As textual contents constitute a big part of the profiles of users, new methods are needed for the exploitation of this content to detect and identify the competence of users. Thus, our main objective is to propose an ML-based model that predicts a competence level of users from their text.

The remainder of this paper is organized as follows: Section 2 discusses the notion of competence in the context of this study. Section 3 presents related literature review. Section 4 describes the corpus used in this study and section 5 and 6 discuss the text analysis and ML methods used respectively. Section 7 and 8 discuss the training and testing phase. Section 9 discusses the evaluation and comparison of the selected ML methods. Section 10 discusses the results. Section 11 describes future works.

2 The Notion of Classification Competence

The users of SNS classify a number of images and discuss the respective classifications. *Competence* is defined as the ability to identify the animals appearing in the images and classifying them into species correctly. For example, the animal in an image looks like a Weasel of some kind, one user might classify it as a Mongoose, whereas another user might confuse it with a Zorilla as the two species have some characteristics in common.

To assess the expert level of the users, a majority vote scheme has been applied to the classifications of each image. That means that, each image is shown to multiple volunteer users then majority votes are taken as a accurate classification of the animal appeared on the image. Depending on the number of correctly classified images carried out by individual

[1] https://www.snapshotserengeti.org
[2] https://www.mico-project.eu

users, a weight is assigned to each user as an overall performance using a rating scale from 0 (the least competent users) to 1 (the most competent users) is used.

Obviously, relying on the majority votes as a ground truth has a number of problems, for example an image might contain an animal which is a bit hard to be discerned by non-expert users, but it probably receive majority votes for inaccurate classification than expert users. Thus, we need to come up with a better apprach, for example by adapting previously proposed algrorithms [5]. Authors in [5], carried out a quite detailed analysis to develope a *weighted majority voting* method for combining users annotations into an overall classification. A mechanism has also been devised for handling *blank classifications*, where some very blurry images are reported as blank by some users, but those images might contain animials.

3 Related works

The most related works in the area of competence analysis from textual contents include [3, 6]. In [3], the use of ML and NLP methods to evaluate the competence of medical students from their clinical portfolio has been discussed. Specific competence goals have also been defined according to the competence-based curriculum practiced in the medical schools in USA. That allows them to identify the potential features associated with competence and extract those features from students notes. Moreover, they make use of available resources such as unified medical language system (UMLS) and knowledgeMap concept indexer (KMCI), that make the modeling a bit easier. On the contrary, we do not have such domain-specific resources. Thus, we make use of methods that automatically extract useful patterns representing competence from annotated training data without using external knowledge sources, e.g bag-of-words models. However, then we end up with a very large number of features, as each word in the corpus is a feature. Compared to other supervised learning methods, DT, NB and KNN are better alternatives due to their efficiency for data in high dimensional space.

A preliminary study has been carried out in [3] to apply ML-based methods to identify student experiences in different competence medical domains. They use a well-defined set of competence goals recommended by the two national accreditation bodies in USA including the Accreditation Council for Graduate Medical Education and the American Association of Medical Colleges. The proposed approach utilizes the medical domain specific models such as UMLS and KMCI to detect biomedical concepts from students notes . In their experiments, they trained three ML-based classifiers, NB, SVM and LR on an annotated corpus consists of 399 clinical notes. Their result shows that the performance of the ML methods vary across the competence domains identified for medical students.

A comprehensive survey about the existing state of the art approaches for automatic essay scoring has been presented in [6]. Various learning techniques used in earlier studies have also been discussed in the survey, such as classification and regression. Regression-based approaches include support vector machine (SVM). Where as, the classification-based approach includes NB and KNN.

As argued in Section 1, we apply the selected ML methods to the pre-processed (with natural language processing (NLP) tools) and annotated dataset of the SNS. We train the selected ML methods on labeled text and evaluate their prediction accuracy through cross-validation technique. We also experiment with strategies that could potentially enhance the prediction performances of the chosen ML methods. These strategies are feature engineering, and scale calibration. Then we run the corresponding comparisons.

4 Corpus Description

We use a corpus of comments collected from the SNS forum for both training and testing ML based models. All comments written by individual user have been aggregated into a document so that each document is labeled with *competence values*, ranging from 0.00 to 1.00. The main idea is to predicte the *competence level* for the new users based on their comments using learned models. The corpus contains the comments generated by a total of 5,243 distinct users. We have mainly two types of experimental settings baseline and calibrated setting. In a baseline setting we apply fine-grained competence labels to divide the users into 5 catagories *very incompetent, incompetent, average, competent, very comptent* based on their competence values, ranging [0.00, 0.2], (0.20, 0.40], (0.40, 0.60], (0.60, 0.80], and (0.80, 1.00] respectively. Most of the users fall into *very competent, competent* catagories. To reduce this *class-imbalance* problem and improve the performance of the models, we attempted to calibrate the *competence scale* to have three catagories *competent, average, incompetent*, ranging [0.00, 0.33], (0.33, 0.67] and (0.67, 1.00] respectively. We attempted to analyze the behaviour of the selected models by doing so and got a dramatic improvement of the prediction accuracy.

5 Text Analysis

The NLP tasks for the text corpus have been performed by the Rapidminer tool [3], which is an open source software for data mining. Rapidminer provides several text analysis modules and ML algorithms. Rapidminer is a plausible choice to approach our problem because it supports the extrac-

[3] https://rapidminer.com

tion of bag-of-word features from a raw unstractured text corpus. Moreover, Rapidminer provides Java API support for the integration of the resulting ML-based models into MICO, to which we implement text analysis components. We run the following main text processing operations on the text corpus to produce a featured dataset, so that it can be used to train the selected ML models.

Tokenization splits the comments posted by each user into a sequence of tokens, a token for example, might be a word. There are several ways of doing that by using regular expressions, specially to control tokens containing non-standard characters, but to keep the originality of the contents we used the default setting of the Rapidminer tokenizer module. Avoiding any kind of exclusion of such characters also potentially contribute to the ML models to learn the actual fact of the contents.

Stemming takes the word tokens returned during the tokenization phase and generate a morphological base form of the words by stripping the word suffixes.

Generating n-grams an n-gram is a sequence of tokens of length n. Here we generated bi-grams (n=2) and tri-grams (n=3) to use them as additional features to the baseline bag-of-word feature set. Since all possible sequences of tokens have to be generated for each document and since they also turn out to be a part of a feature set, it becomes computationally expensive. Due to this problem, we could only apply the bi-gram and tri-feature to the NB model.

Extraction of number of tokens returns the total number of tokens in each document and is another important feature intended to be included in characterizing the text written by the users. All types of tokens have been counted regardless of their lengths.

Extraction of aggregate tokens length it is a computation of the aggregate length of all tokens in a text.

6 Method Description

We give a formal and brief description for the three ML models used in this study, naive bayes [4], decision trees [8] and K-nearest neighbour [14] . In addition to their bag-of-words features support, we chose these models because they are easy to understand and interpret, and implement.

6.1 Naive Bayes

Naive Bayes (NB) is a probabilistic classifier and applied to several text classification problems [6]. Once trained with a corpus of documents, the NB model returns the most probable class for the input text based on the Bayes rule of conditional probability. First, the text (a document) needs to be defined and represented with a set of features and we also assume that T is a set of training samples. Then the NB takes a feature vector $\overrightarrow{d} = (f_1, \ldots, f_n)$ of the document and applies the following equation to predict the most likely class:

$$\underset{C}{\operatorname{argmax}}\, P(C|\overrightarrow{d}) \tag{1}$$

$$P(C|\overrightarrow{d}) = \frac{P(f_1, \ldots, f_n|C)P(C)}{P(f_1, \ldots, f_n)}. \tag{2}$$

Here the term $P(C|\overrightarrow{d})$ is a probability of $\overrightarrow{d}$ being in a class C, defined as:

$$P(C|\overrightarrow{d}) = \frac{P(C)\prod_{i=1}^{n} P(f_i/C)}{P(f_1, \ldots, f_n)}. \tag{3}$$

Here the term $P(C)$ is a prior probability of a class C and (f_i/C) is a conditional probability of f_i given a class C. Since the $P(f_1, \ldots, f_n)$ is the same for all classes. Then, the above equation can be reduced to:

$$P(C|\overrightarrow{d}) = P(C)\prod_{i=1}^{n} P(f_i/C) \tag{4}$$

A probablity P over T is estimated based on word/token and class counting as follows:

$$P(C) = \frac{count(C)}{|T|}. \tag{5}$$

$$P(f_i/C) = \frac{count(f_i, C)}{TC}. \tag{6}$$

Here $count(C)$ returns the number of times that the class C is seen in T, and $|T|$ is the total number of samples in the training corpus, TC is the Total number of (word/token) in a class C. In a bag-of-words model each feature f_i for i=1...n, represents a word/token, therefore $count(f_i, C)$ returns the number of times the word/token f_i seen in the class C. To avoid *zero probabliities*, laplace correction (add-one smoothing) has been used. That is a commonly used parameter smoothing technique which adds one to each count.

6.2 Decision trees

Decision trees (DT) is extensively used in a wide range of NLP applications for building tree structured predictive models. Decision trees built by the DT algorithm consist of a root node, which represents the most discriminatory feature in the training feature set, edges, that represent answers to questions asked by internal nodes, and leaf nodes that represent decisions [8]. To split training samples (T) with n number of classes of the form, $(f_1, \ldots, f_n, C)$ into subtrees, the DT algorithm computes the Entropy (H), which is a measure of homogeneity of T, and the Information Gain (IG), which is a measure of a decrease in H.

Here are the equations for H and IG respectively:

$$H(T) = -\sum_{i=1}^{n} P(C_i)\log_2 P(C_i), \tag{7}$$

where the term $P(C)$ is a probability of a class C_i. The IG for any f_i in a feature set characterized the T defined as:

$$IG(T, f_i) = H(T) - \sum_{x \in X} P(x) \sum_{i=1}^{n} P(C_i|x) \log_2 P(C_i|x), \tag{8}$$

where X is a set of values of the feature f_i in T, and the term $P(x)$ is a probability (see equations 4 and 5 for its estimation) of the value $x \in X$.

During a decision tree construction, the feature yeilding the highest IG taken by the DT algorithm to split the samples recursively.

7 K-Nearest Neighbour

K-Nearest Neighbour (KNN) is a non parametric classifier. In the KNN algorithm, K represents the number of samples in a training set that are closest to an input sample. Those samples belong to the class predicted by the algorithm. The nearest neighbours to the input samples are obtained by using, for example, Euclidean distance. KNN has been used in many applications such as search engines [7], and pattern matching [14].

Euclidean distance between the two feature vectors, $({f_1}^1, \ldots, {f_n}^1)$ and $({f_1}^2, \ldots, {f_n}^2)$ representing two documents $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ respectively can be estimated as:

$$D(\overrightarrow{d_1}, \overrightarrow{d_2}) = \sqrt{\sum\nolimits_{i=1}^{n} ({f_i}^1 - {f_i}^2)^2}. \tag{9}$$

8 Training and Testing

To train and test the models, we have randomly split up the whole corpus composed of 5,242 samples and 10,062 features, into 70% training and 30% test set. We make use of particulary a *shuffled sampling*, where samples are chosen with random orders. Before the split, the target lable *accuracy* has been discretized from a numeric type into a nominal type to meet the requirement posed by the ML algorithms implementaion in the Rapidminer ML environment. During the traing phase, the models parameters have been optimized to reduce model *model over-fitting* through a held-out data set.

9 Comparision Across Models

The following commonly used standard metrics [15] (i.e., their values range between 0 and 1) have been used to measure the performance of the models:

Accuracy (A) as it is defined in equation 10, that tells how many of the documents (here, the single document represent the text written by each user) are correctly classified out of the test set.

Precision (P) as it is defined in equation 11, that indicates the sensitivity of the models towards true predictions.

Recall (R) as it is defined in equation 12, it shows that how the models performs for each class on the basis of the size of their test set.

F-measure is a harmonic mean of P and R. For a binary classification its estimation is stright-forward. For multi-class problems, there are two common approaches, micro-averaging and macro-averaging of F-measure. Micro-averaging takes the global values of P and R for the F-measure estimation, whereas the macro-averaging takes the local values of P and R. In micro-averaging the F-measure has the same value as accuracy unless a bias is estimated which is mostly applied to a cross-model analysis. Because of that, we used the macro-averaging to compare the models investigated in this study.

Given that each instance represents a text written by a user and every C_i for i=1...N is a subset of a test set T, where N is the number of classes, we define the following equations for the metrics, A, P_i, R_i and $F\text{-}measure_i$ respectively:

$$A = \frac{\sum_{i=1}^{N} TP_i}{|T|} \tag{10}$$

$$P_i = \frac{TP_i}{(TP_i + FP_i)} \tag{11}$$

$$R_i = \frac{TP_i}{(TP_i + FN_i)} \tag{12}$$

$$F\text{-}measure_i = \frac{2P_iR_i}{(P_i + R_i)} \tag{13}$$

$$F\text{-}measure = \frac{\sum_{i=1}^{N} F\text{-}measure_i}{N} \tag{14}$$

Where:

- TP_i(true positive) is the number of instances accurately predicted to class C_i
- FP_i(false positive) is the number of instances wrongly predicted to class C_i
- FN_i(false negative) is the number of instances belong to class C_i, but not accurately predicted to that class

Table1Cross-validation and comparision results

Models	Accuracy (%)		F-measure (%)	
	Baseline Features	Added Features	Baseline Features	Added Features
DT	79.34	80.04	19.51	32.62
NB	48.38	48.38	21.71	21.71
KNN	63.19	69.42	21.74	33.79
Hybrid	73.74	74.00	-	-
After calibration				
DT	93.39	94.09	34.42	52.47
NB	68.28	68.28	33.18	33.18
KNN	88.94	91.61	33.60	45.58
Hybrid	91.86	93.26	-	-

10 Discussion

The baseline validation results shown in Table 1 tell us that the DT outperforms the other two models, NB, and KNN, by 16.00% and 15.00% of accuracy, respectively. However, the DT has got the lowest value of F-measure with the baseline features, due to its smallest recall value. We also attempted to build a hybrid model that combines the three models (DT, NB and KNN). However the hybrid model has relatively a poor performance regarding F-measure, given that it has much better accuracy value than NB. These figures indicate the possibility of predicting user classification competence using the selected ML models based on their comments to a certain degree of effectiveness. These models have been studied with other supervised learning models in [10, 11] for text classification problem, it has been shown that DT and KNN outperform NB regarding both accuracy and F-measure metrics. One of the possible reasons for the poor performance of the NB model may be its strong independent interaction assumption between features, words/tokens which act as unigram features of the model [4].

We added more features (see the description below) to enhance the performance of the models and we achieved a substantial prediction accuracy increase with only the DT and KNN models. The performance of the NB has been improved by exclusively adding a bigram feature, however, we do not consider that improvement in the comparison with other models to avoid unfair comparisons (see the details below). We have also achieved even more increase in performance regarding accuracy and F-measure with a calibration technique applied to all the three models.

We tried to observe and analyze evaluation results in various possible conditions. Having those conditions provide us different perspectives of the problem being studied. That is also important to ensure the reliability of our experiments through observable consistencies of invariant parameters in all conditions. The first condition in our experimental settings is the models training with five different classes and baseline bag-of-words features. These classess represent five catagories of users in the scale of from the least competent users to the most competent ones. The second condition followed by adding more features such as a number of classification (NoC), a number of tokens (NoT), and aggregate token length (LoT) to the the models trained in the first condition. Here, the combined effect of the NoT and LoT has been studied independently as well as the NoC. In the third condition, the models have been trained to have three different classes to capture three categories of users, namely, less competent, competent, and high competent. Then we applied the first two conditions as the subconditions. The next paragraph discusses how the added features impact the performance of the models.

NoC represents the total number of images classified by each user of SNS. This feature has improved the accuracy of DT (from 79.34% to 80.04%) and KNN(63.19% 68.21%), which indicates their sensitivity to the numerical features as compared to the NB model. Specially, the DT used this feature as a root node during the construction of its decision tree, which means the NoC has been taken as the best predicator even from the basic features. Obviously, it is natural to assume that a user with a high number of classifications to fall in the *very competent* category, despite it is not always true. Moreover, the corpus used in this study also reflect this fact as well, for example, there are users who made classifications of more than 18,000 images, and they are in the *very competent* category. This shows that how experiences affect the classification proficiency of the users.

NoT and LoT these features represent the size of the comments posted by the users in terms of tokens/words. We are interested in observing their combinational effect on the prediction of competence, they are closely related and assumed to be the good indicators of competence as the most competent users tend to write long comments. Unfortunately, these features have no any effect on neither of the models.

Bi-gram represents a sequence of two words. This feature has dramatically improved on (from 48.38% to 64.40%) the accuracy of the NB model. Having the more contextual information in texts always improves the efficiency and accuracy of its classification. Generally, it is also an evident that n-gram features have impact on several text classification applications. Due to the more memory requirement, we could not see the effect of the bigram feature in the DT and KNN models. One possible approach to face this challenge is reduce the size of the training data to meet, but which might cause unfair comparison due to a different setting.

We attempted to analyze the performance of the selected ML models and make a generalization with a limited number of experimental conditions. However, still different results might be obtained and have new perspectives to the problem if we had approached it differently. For example, as we mentioned in section 2, we followed a heuristic approach to set the ground truth for computing user *competence*, but there are other possible ways to try that could potentially produce better results.

We assumed that a written text is probably a good indicator about the classification competence of users, but this assumption does not hold for some cases, for example some competent users (given that they made *accurate classifications*) may not be interested or have time to discuss their classifications as much as incompetent users. In this case, our models could fail to detect the real competence of such users.

On the one hand, leveraging the NLP components available in the Rapidminer tool has been useful for general text analysis and bag-of-words feature extraction in our study. But on the other hand, considering additional tools dealling with more advanced aspects of text, such as noisiness [12] could potentially improve our results. Noisiness is a problem commonly associated with the text obatined from social media as well as citizen science project media, that makes a parsing a bit hard.

10.1 Calibration Technique

As it is noted from Table 1, we have a class-imbalanced data problem. This causes bias to the test dataset to be classified into these classes regardless of their actual classes, and there by severely affects the prediction accuracy of the models. So, one of the possible approaches to this problem is to take the distribution of the target feature into account and divide the corpus into three classes based on equally sized and partitioned ranges of the accuracy values. By doing so, we achieved pushing the accuracy limit in the baseline models to 94.09%.

11 Conclusion and Future Work

In this study, we achieved three main goals, applying the selected ML models DT, KNN and NB, effectively to learn a user competence from the text obtained from the SNS posts, evaluating and comparing the performance (regarding accuracy, precision, recall and F-measure) of these models and improving the baseline results through additional features and scale calibration. Regarding accuracy, DT outperforms NB and KNN by 16.00%, and 15.00% respectively. Regarding F-measure, K-NN outperforms NB and DT by 12.08% and 1.17%, respectively.

The learned models in this study can be applied to other citizen science projects such as the galaxy project[4] supported by an online platform where images of galaxies are posted to be classified by volunteer users. Moreover, the model performance improving techniques proven to be effective in this study could be useful in other related areas such as text classification problems.

Our next steps to further improve our resuts are to consider and experiment with other ML and Ontology based models such as SVM and Neural Nets, and large dataset. We will also attempt to work on implementing a new ground truth for competence estimation. Then the resulting model has been intended to be integrated into a cross-media analysis framework MICO.

12 acknowledgements

We acknowledge the financial support from the EU FP7 MICO project. We also thank the Zooniverse people for providing the dataset used in this study.

References

[1] Agarwal Alekh and Bhattacharyya Pushpak. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In *Proceedings of the International Conference on Natural Language Processing (ICON)*, pages 79–86, 2005.

[2] Pang Bo, Lee Lillian, and Vaithyanathan Shivakumar. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, 2002.

[3] Y. Chen, J. Wrenn, H. Xu, A. Spickard, R. Habermann, J. Powers, and J. Denny. Automated assessment of medical students' clinical exposures according to AAMC geriatric competencies. In *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 375–384, 2014.

[4] D. Lewis David. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 4–15, 1998.

[5] Hines Greg, Swanson Alexandra, Kosmala Margaret, and Lintott Chris. Aggregating user input in ecology citizen science projects. In *AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3975–3980, 2015.

[6] Chen Hongbo and He Ben. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, 2013.

[7] S Ján and Návrat Pavol. *Full Text Search Engine as Scalable k-Nearest Neighbor Recommendation System*. Springer, Berlin Heidelberg, 2010.

[8] Breiman Leo, Friedman Jerome, J. Stone Charles, and Olshen R.A. *Classification and Regression Trees*. Wadsworth, Monterey, CA, 1984.

[9] Entezari-Maleki R, Rezaei A, , and Minaei-Bidgoli B. Comparison of classification methods based on the type of attributes and sample size. *Journal of Convergence Information Technology (JCIT)*, 4(3):94–102, 2009.

[10] Entezari-Maleki R, Rezaei A, and Minaei-Bidgoli B. Text classification using keyword extraction technique. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(12):734–740, 2013.

[11] Nomponkrang Thanyarat and Sanrach Charun. The comparison of algorithms for thai-sentence classification. *International Journal of Information and Education Technology*, 6(10):801–808, 2016.

[12] Baldwin Timothy, Cook Paul, Lui Marco, MacKinlay Andrew, and Wang. Li. How noisy social media text, how diffrnt social media sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 356–364, 2013.

[13] Y. Woldemariam. Sentiment analysis in a cross-media analysis framework. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pages 1–5, 2016.

[14] Wu Y, Ianakiev K, and Govindaraju V. Improved k-nearest neighbor classification. *Pattern Recognit*, 35:2311–2318, 2002.

[15] Y Yang and X Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 1996.

[4] https://www.galaxyzoo.org

III

Predicting User Competence from Linguistic Data

Yonas Demeke Woldemariam
Dep. Computing Science
Umeå University
Sweden
yonasd@cs.umu.se

Suna Bensch
Dep. Computing Science
Umeå University
Sweden
suna@cs.umu.se

Henrik Björklund
Dep. Computing Science
Umeå University
Sweden
henrikb@cs.umu.se

Abstract

We investigate the problem of predicting the competence of users of the crowd-sourcing platform Zooniverse by analyzing their chat texts. Zooniverse is an online platform where objects of different types are displayed to volunteer users to classify. Our research focuses on the Zoonivers Galaxy Zoo project, where users classify the images of galaxies and discuss their classifications in text. We apply natural language processing methods to extract linguistic features including syntactic categories, bag-of-words, and punctuation marks. We trained three supervised machine-learning classifiers on the resulting dataset: k-nearest neighbors, decision trees (with gradient boosting) and naive Bayes. They are evaluated (regarding accuracy and F-measure) with two different but related domain datasets. The performance of the classifiers varies across the feature set configurations designed during the training phase. A challenging part of this research is to compute the competence of the users without ground truth data available. We implemented a tool that estimates the proficiency of users and annotates their text with computed competence. Our evaluation results show that the trained classifier models give results that are significantly better than chance and can be deployed for other crowd-sourcing projects as well.

1 Introduction

The science crowd sourcing platform Zooniverse hosts a large number of different projects where volunteers/users (in this paper, the term “volunteer” is used interchangeably with “user”) help scientists by classifying various kinds of data. In order to make the experience as positive as possible for the volunteers, so that they are more likely to stay on and contribute to the projects, the Zooniverse team is very interested in anything that can help them understand their volunteers better.

In this article, we explore how much the text comments left by volunteers in the chat rooms accompanying the project Galaxy Zoo can help us in determining their level of proficiency or competence in classifying images. Proficiency is only one among many interesting qualities, and the text data is only one tool for measuring it. The output from the machine learning algorithms we use can be combined with other measures to learn more about user proficiency. Here, though, we focus on the following main question: Does the linguistic data from the chats contain useful information about the volunteers, in particular about the quality of their classifications?

The reason for focusing on Galaxy Zoo, rather than one of the many other projects run by Zooniverse, is that it is one of the oldest and largest projects, which means that there is quite a lot of data available – many users, many classifications, many text comments.

There are several challenges that have to be addressed when trying to answer our question. The hardest one is how to measure the quality of users’ classifications. The problem is that there is no ground truth data available. For most of the galaxy photos that volunteers have classified, we do not know the correct answer. No expert in the field has studied and classified them, since the whole point of using volunteers is that the experts do not have the time to do so.

Our approach to this challenge is to use majority votes, i.e., we consider the answer to a question given by the majority of the users to be the correct one. This is by no means an unobjectionable assumption. We describe our approach in more detail and provide some justification for it in Section 3.

Once a quality measure for each user that has also provided sufficiently many textual comments has been computed, we employ three different machine learning algorithms to the data in order to see whether the values can be predicted from text. Each algorithm is tested on six different sets of features of the textual data. The algorithms we use are k-Nearest Neighbors, Naive Bayesian Classification, and Decision Trees (with gradient boosting).

The results achieved are not spectacular, but they show that analysis of the textual data gives a significantly better than chance prediction of the quality of a users classifications. As mention above, this can be combined with other measures to get better predictions.

To investigate how well our methods generalize to other settings we also test them on data from the Zooniverse Snapshot Serengeti project. The results are encouraging in that they are comparable to the results for Galaxy Zoo.

We discuss related work in Section 2, the calculation of majority votes in Section 3, the experimental setup in Section 4, the experimental results in Section 5 and, finally, the discussion in Section 6.

2 Related work

In the literature a users' competence refers to various kinds of competence. Automated essay scoring, for instance, assesses an author's writing competence or capabilities by analyzing the author's text. An author's competence can also refer to competence or expertise in a specific topic that he/she demonstrates by, for example, his/her written argumentation in a chat discussing the topic. An author's competence can also be related to the author's competence in performing a specific task (e.g. classifying galaxy images) and the author's written text about the task performance can be used to investigate whether there exist correlations. We are interested in the correlation between an author's task performance competence (i.e. correct classification of galaxy images) and his/her chat entries, where the text in the chat entries is not necessarily about the task at hand.

Researchers have intensively investigated methods for automated essay scoring by statistical analysis of linguistic features extracted from text. Automated essay scoring is the process of automatically analyzing text and grading it according to some predefined evaluation criteria. In McNamara et al. (2008), for instance, the authors investigate to what degree high- and low-proficiency essays can be predicted by linguistic features including syntactic complexity (e.g. number of words before the main verb). Their results indicate that high-proficiency writers use a more complex syntax in terms of the mean number of higher level constituents per word and the number of words before the main verb, than low-proficiency writers. In addition, the results indicate that high-proficiency writers use words that occur less frequently in language. Chen and He (2013) improve automated essay scoring by incorporating the agreement between human and machine raters. The feature set to indicate essay quality includes lexical, syntactic, and fluency features. The syntactic features include sentence length, the mean number of subclauses in each sentence, the sum of the depth of all nodes in a parse tree as well as the height of the parse tree. In Pérez et al. (2005), students' essays are assessed by combining an algorithm that includes syntactic analysis and latent semantic analysis.

Linguistic features in written text (e.g. chat) have also been used to predict how competent the authors are with respect to learning and understanding discussed chat topics. Dascalu et al. (2014), for instance, assess the competences of chat participants. To this end, they consider the number of characters written by a chat user, speech acts, keywords and the topics. In addition, social factors are taken into account. The authors generate a social network graph that represents participants' behaviors and participants can be characterized as knowledgeable, gregarious or passive. The social network is used to compute metrics such as closeness, graph centrality, betweenness, stress, and eigenvector.

Linguistic features have been used to predict text-specific attributes (e.g. quality of text) as well as author-specific attributes. In Kucukyilmaz et al. (2008) the authors predict user-specific and message-specific attributes with supervised classification techniques for extracting information from chat messages. User-specific attributes include, for example, gender, age, educational background, income, nationality, profession, psychological status, or race. In Kucukyilmaz et al.

(2008) a term-based approach is used to investigate the user and message attributes in the context of vocabulary use and a style-based approach is used to investigate the chat messages according to the variations in the authors' writing styles.

Another kind of author-specific attribute is the self-confidence of an author. In Fu et al. (2017) the authors investigate how confidence and competence of discussion participants effect the dynamics and outcomes of group discussions. The results show that more confident participants have a larger impact on the group's decision and that the language they use is more predictive of their confidence level than of their competence level. The authors use bag of words, number of introduced ideas, use of hedges (i.e. expressions of uncertainty or lack of commitment) and expressions of agreement as indicators for confidence.

Berry and Broadbent (1984) investigate the relationship between task performance and the explicit and reportable knowledge about the task performance (i.e. concurrent verbalization). The results indicate that practice significantly improves task performance but has no effect on the ability to answer related questions. Verbal instructions of how to do the task significantly improves the ability to answer questions but has no effect on task performance. Verbal instructions combined with concurrent verbalization does lead to a significant improvement in task performance, whereas verbalization alone has no effect on task performance or question answering. The authors Berry and Broadbent (1984) use statistical comparisons of questionnaires.

In Chen et al. (2014), the authors use machine learning techniques (e.g. logistic regression, SVM) to assesss medical students' competencies in six geriatric competency domains (i.e. medication management, cognitive and behavioral disorders, falls, self-care capacity, palliative care, hospital care for elders). The medical students' clinical notes are analyzed and the used linguistic features include bag of words, concepts, negation and semantic type. The authors also use non-linguistic features such as the number of clinical notes for the competence assessment.

3 Computing majority votes

Schwamb et al. (2005) assess how competently a volunteer can identify planetary transits in images. This is done within the Planet Hunter project[1] which is a crowd sourcing project for which volunteers classify planet images. A decision tree helps volunteers in identifying light curves in the images and the volunteers then mark transit features visible in the light curve which results in a so-called transit box. The classifications are stored in a database and for each entry question in the decision tree, the time stamp, user identification, light curve identifier, and response are stored. In addition, the position of the transit box center, its width and height are stored. As a gold standard synthetic transit light curves are used (i.e. labelled images) where these synthetic transits are mixed into the images that are not labelled for the volunteers to classify. In order to identify the most competent volunteers a weight is assigned based on their tendency to agree with the majority opinion and in case they classified synthetic light curves on their performance of identifying transit events. The user weights' are assigned in two stages. First, all users start out equal and then the results of identifying the synthetic light curves are used to obtain an initial weighting. For every synthetic light curve and volunteer classifier it is evaluated how well the user identified the transit events. If a volunteer identified transits correctly her weight is increased and if a volunteer did not mark any synthetic transits (transit box) her weight is decreased. For all the volunteers who classified non-synthetic images the competence evaluation is based on majority opinion. A volunteer's weight increases if the volunteer is in line with the majority weighted vote and is decreased if the volunteer is not in line with the majority opinion.

One of the major obstacles to our investigation was that there is no gold standard data available for the Galaxy Zoo subjects. (A subject is the Zooniverse term for a unit that is presented to volunteers for classification. In the case of Galaxy Zoo, this is one photo taken by a telescope.) In other words, we do not know what the correct classification for the images are. This, in turn, means that there is no way of computing a gold standard for the competence level of the volunteers, since we cannot with certainty determine whether they have classified an image correctly or not.

For these reasons, we had to find a way of estimating the competence levels. How best to do this is not at all obvious. The one approach that

[1] `planethunters.org`

we have judged possible is to use majority votes, in essence trusting that most classifications are correct. This assumption is at least in part justified by the fact that if it were not true, the whole Galaxy Zoo project would be pointless. The lack of gold standard data prevented us from using a more sophisticated model, where the volunteers performance on classification tasks with a known answer is used as an initial weighting, which is then reinforced by considering majorities on other classification tasks. Such an approach has been used in Planet Hunters, another Zooniverse project (Schwamb et al. (2005)).

In order to explain our approach in detail, we must first say something about the structure of the classification tasks the volunteers are presented with. Each subject is associated with a decision tree based flow chart of questions. The exact chart varies slightly depending on which sub-project of Galaxy Zoo the subject belongs to, but generally, the volunteers are asked three to five questions for each subject, where each of the questions following the first one depends on the answers to the previous questions. Since most subjects in the database have between 10 and 20 classifications, we determined that computing the majority votes for a whole subject classification, including all the questions from the flow chart, would not be advisable, since the answers to the questions after the first one vary to a surprising degree. We thus made the pragmatic decision to only consider the answers to the first question for each subject.

When a volunteer is presented with a subject, the first question, irrespective of which sub-project the subject belongs to, is whether the object in the middle of the photo is a smooth galaxy, a galaxy with features (a disc, spiral arms, etc.), or looks like a star or some other artifact. There are thus three possible answers to the first question. The first step was therefore to calculate, for each subject, how many volunteers had given answers 1, 2, and 3, respectively. In order to have a reasonable amount of data for each subject, we disregard subjects with fewer than 10 classifications.

The next step was computing a competence value for each volunteer that had done at least 10 classifications. Here, we again had some design choices to make. The easiest approach would have been to simply say that for each subject, the correct answer is the one that has gotten the most votes, and then count, for each volunteer, how many times they had given the correct answer and dividing this number by the number of classifications the volunteer had performed. This approach, however, has serious drawbacks. In the data set, it is not uncommon to find subjects where no answer has a clear majority. Consider a case where answer 1 has 12 votes, answer 2 has 10, and answer 3 has 4. Here, it is not clear which answer is actually correct, and it would be bad to give a "full score" to the volunteers that had given answer 1 and no points at all to those that had given answer 2.

Instead, we decided to go by the assumption that the more other volunteers agree with you, the more reasonable your answer is. We thus calculated the competence score for a volunteer as follows. For each subject that the volunteer has classified, we divide the number of votes that agree with the volunteer by the total number of votes, getting a number in the interval $[0, 1]$. The score for the volunteer is then the average of these numbers over all subjects the volunteer has classified. This approach has the benefit of "punishing" a volunteer more severely for an incorrect answer to an "easy" question, where most other volunteers have voted for another answer, while being lenient towards false answers to "hard" questions. On the downside, the users answering the hard questions correctly, get less credit for this than they deserve.

4 Experimental setup

4.1 Text Analysis and Feature Extraction

We extracted text comments written by 7,839 volunteer. We only targeted those users who classified at least 10 subjects and discussed at least one of their classifications in chat text. The users were divided into three categories of equal size based on their computed competence levels on a scale ranging from 0 to 1: low ($[0, 0.52]$), medium ($(0.52, 0.59]$) and high ($(0.59, 1]$). Having an equal number of users in each category helps to achieve balanced data and in eliminating bias during the machine learning phase. The raw data was obtained from Zooniverse Galaxy Zoo as a database dump. The entire text data contains around 26,617 sentences with average sentence length of 5.02. We extracted three types of linguistic features out of the text data: bag-of-words, syntactic and punctuation marks. The number of classifications is also included in each feature set as special feature or meta data.

4.1.1 Syntactic feature set

To extract syntactic features the Stanford probabilistic context-free grammar (PCFG) parser was used Klein and Manning (2003). These features provide a lot of information about the complexity of the syntactic structures used by the volonteers. For each syntactic category, we made a correlation analysis with classification competence. To this end, we implemented a Java-based program that reads user texts from the database stored on the Mongodb server running on a local machine and makes use of the PCFG model to construct a syntactic parse or phrase-structured tree for the texts. The program counts the frequency of syntactic categories/constituent tags occurring within the tree and then annotates the text with these tag count information.

The non-leaf nodes in the resulting tree has three major syntactical categories: lexical categories, functional categories and phrasal categories. The lexical categories are the part-of-speech tags of the leaf nodes that represent content words that make up the parsed text, for example, NN (noun), JJ (adjective), VB (verb), etc. As the Stanford parser has been trained on the Penn Treebank, we use the part-of-speech tags and their notations used in the tree bank to label the non-leaf nodes as well as to identify categories. The functional categories contain items responsible for linking syntactic units, for example, DT (determiner), IN (preposition), MD (modal), etc. The phrasal categories represent different type of phrases within a sentence for which the parse tree is built, for example, NP (noun phrase), VP (verb phrase) and AP (adjective phrase), etc. In the syntactic feature set there are 66 numerical attributes representing the frequency count of syntactic categories.

We attempted to analyze the correlation between the syntactic categories count with computed competence values by looking at the correlation coefficient(CC) calculated for each syntactic category as summarized and shown in Figure 1. The calculated CC values range $[-0.05, 0.04]$, statistically speaking these values do not show that there is a strong relationship. The Figure basically shows three types of relationships between the syntactic categories and competence according to the observed CC values: the first type of relationship is exhibited by the categories laid over the left-hand side of the X-axis such as JJR (adjective, comparative), PRP$ (possessive pronoun) and JJS (adjective, superlative) they are negatively correlated with competence, those concentrated around the center such as S (simple declarative clause), PRT (particle) and WP$ (possessive wh-pronoun), do not seem to have a correlation with competence and the third type of relationship is exhibited by the categories close to the right-hand side of the X-axis such as PRP (personal pronoun), SQ (inverted yes/no question) and SBARQ (direct question introduced by a wh-word).

4.1.2 Punctuation mark feature set

We also extracted the frequency count of punctuation marks including question mark, period, and exclamation mark. Special characters such as # and @ were also included. We also tried to perform a correctional analysis between each feature in the set with competence as we did for the syntactic feature set and we got quite similar results in terms of the strength of their relationship. In the punctuation mark feature set there are 7 numerical attributes, that correspond to the selected punctuation marks.

4.1.3 Bag-of-Words feature set

We used the text analysis package of Rapidminer[2] and text-processing Java libraries to extract the Bag-of-Words (BoW) and punctuation marks features respectively. The text analysis involves splitting text into sentences, each sentence is further split into words followed by stemming and part-of-speech tagging. In the Bag-of-Words feature set there are 19,689 attributes excluding the target (label) attribute, i.e competence. Each attribute has a numerical value that represents the frequency count of any token in a text.

By taking both an individual feature set and combination of them, we came up with 6 feature set configurations: Bag-of-Words (BoW),punctuation marks (Pun), punctuation marks with Bag-of-Words (Pun+BoW), syntactic, syntactic with Bag-of-Words (Syn+BoW), and the combination of BoW, punctuation mark and syntactic (BoW+Pun+Syn).

4.2 Training, Validation and Evaluation

We trained and evaluated three machine learning classifiers: Decision Trees (DT) with gradient boosting, Naive Bayes (NB) and k-Nearest Neighbor (KNN). These three methods were also used in

[2] rapidminer.com

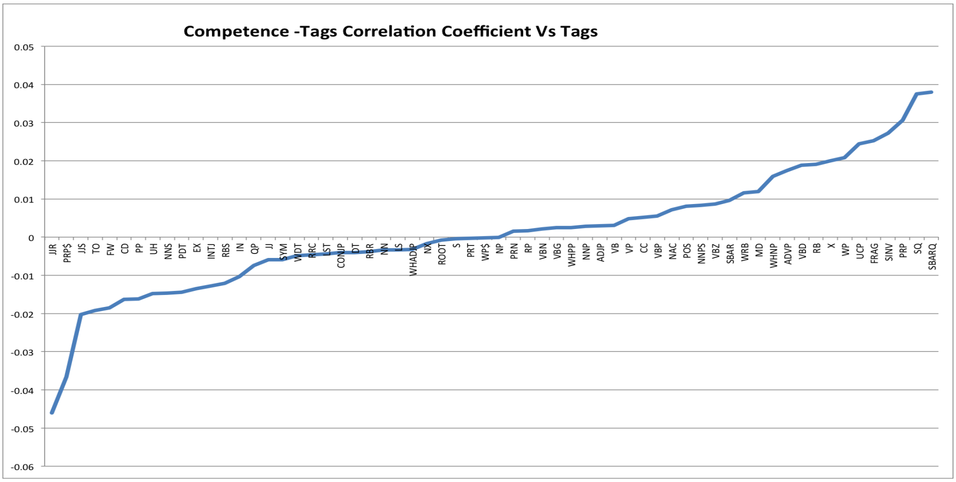


Figure 1: The correlation between frequency of the extracted syntactic categories and computed competence values

a previous study Woldemariam (2017) using Snapshot Serengeti data (another Zooniverse project). As the implementation of these classifiers is available in Rapidminer Studio, we trained them on the Galaxy Zoo data set after configuring the model parameters associated with each classifier.

We adopted the best practices of the machine learning life cycle that includes randomly sampling and dividing the data into a training set, a validation (development) set and a test (evaluation) set, deciding the size of each set and balancing the proportion of examples in each class of users. According to this, the classifiers are trained on 80% of the entire text corpus with the selected feature sets. The remaining 20% is used to evaluate the trained models. We set aside 10% of the training set as a development data set to optimize model parameters.

4.2.1 Training

The classifiers were trained with the different feature sets. The feature sets are applied for each classifier as shown and denoted in the Table 1, first, Bag-of-Words (BoW), second, punctuation marks (Pun), third, punctuation marks with Bag-of-Words (Pun+BoW), fourth, syntactic, fifth, syntactic with Bag-of-Words (Syn+BoW), and sixth, the combination of BoW, punctuation mark and syntactic (BoW+Pun+Syn). Each classifier is trained 6 times with these 6 feature set configurations. Thus, in total, 18 (3*6) classifiers models are produced for the subsequent validation phase. The training set contains texts from 6,262 unique users.

4.2.2 Validation

As a part of the training phase, we attempted to answer whether the trained classifiers are statistically significant before we evaluate them. We performed a null-hypothesis (H_0) test, aiming at checking that the prediction made by the models is not likely just by random chance. In the null-hypothesis we assume that the mean accuracy value before and after testing the models is the same. However, in principle any effective model must have a greater mean accuracy after the testing and reject H_0.

In statistics there are different ways of testing the null hypothesis and the most widely used approach for machine-learning problems associated with models significance test is a T-test. Basically, there are two important parameters in the T-test, a t-value and a p-value. The t-value indicates that how far the mean of the test sample is from the known mean (μ_0), for example, the accuracy mean before testing a model, depends on the size(n), mean ($\bar{x}$) and the standard deviation(s) of a test sample as shown in the Equation 1. The p-value shows how likely the two means are to be equal. Once the t-value is calculated, the p-value can be obtained from a T-table by using degrees of freedom (df).

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \quad (1)$$

So, we performed the t-test for each model with the development set. We found that all the models scored a p-value below 0.001.

4.2.3 Evaluation

The models were evaluated with two equal size test sets by using accuracy and F-measure metrics. The first set is from the same domain as the training set, and the second one is from the Zooniverse Snapshot Serengeti forum discussion posts.

To be able to use the Snapshot Serengeti data, we had to overcome the mismatch of the intervals of the competence scales of the two domains. We had to use a strategy that allows adapting the way that the competence scale for the Galaxy Zoo is divided to label its users to the Snapshot Serengeti users. In Woldemariam (2017), there are two scales used to divide the Snapshot Serengeti users, the first scale divides the user into three groups (Low, Medium and High) and the calibrated scale divides the users into five groups (very Low, Low, Medium, High, very High). Thus, we decided to use the first scale, as it is closer to the Galaxy Zoo scale in terms of the number of divisions, though the intervals between the groups are not exactly the same.

5 Results

The results of the trained classifiers on the test sets are summarized in Table 1. We consider two performance metrics: accuracy and F-measure. To calculate accuracy we take the fraction of true positive and true negative instances (correctly classified instances) among the test instances, while the overall F-measure is computed by macro-averaging the F-measure values over classes. That means the harmonic mean of precision and recall of each class, i.e. the local F-measure of each class, is calculated, then we take the average value over classes as an overall F-measure.

The first thing to notice is that the accuracy scores are low. Since there are three classes in our data (Low, Medium, and High), a random classifier would be expected to have an accuracy of 33%. In our tests, the best classifiers achieve an accuracy of just over 40%. There are, however, reasons why this is not as negative a result as it might seem. First, we work with relatively little data, since most Galaxy Zoo users do not write comments, and no gold standard data is available. There is therefore reason to hope that the approach would yield better results in similar settings, but where more data is available. Second, for the intended use case, Zooniverse, any result that is statistically certain to be better than random is useful. Zooniverse needs a better understanding of their volunteers, both when evaluating the results from classification tasks and in order to learn how to encourage and educate the volunteers. Our classification methods can be combined with other user data to generate such knowledge.

Another interesting aspect is that the results for Snapshot Serengeti are not significantly worse than those for Galaxy Zoo, which indicates that the approach generalizes and can be deployed for other projects as well.

Analyzing the data in more detail, the k-Nearest Neighbors (KNN) classifier performs best overall and in particular when syntax is not involved. When using syntax, it is slightly worse and is sometimes outperformed by the Decision Trees (DT) classifier. It is also interesting that on the Galaxy Zoo data, the best performance (KNN on BoW and PunMM and DT on Syn) are seen when classifiers use only one of the three feature sets. Combining the sets seem to muddy the waters. A partial explanation for this could be that BoW has so many more features than the other two sets.

We also note that the performance of DT and KNN are so similar that we cannot, based on our tests, confidently say that one is a better choice than the other for this task.

The Naive Bayesian (NB) classifiers generally performed the poorest. One potential reason for this is that KNN and DT have flexible model parameters, such as k for KNN and the number and depth of the trees for DT. These values were noted to greatly impact the prediction accuracy during the validation phase. For example, by varying the value of k of KNN model, we achieved about 5% increase in accuracy. Varying the values of the parameters of the kernel-based NB did not help very much in the improvement of its performance.

We also observe that Punctuation mark (PunM) feature set gives the best accuracy value of 40.32% and F-measure value of 40.05%, in this case the Galaxy Zoo test set is used. Generally, according to the evaluation and comparison performed on this test set, the feature sets or their combinations

Table 1: Models Evaluation and Comparison Results, the **All(3)** column is equivalent with BoW+PunMM+Syn

Metric	Domain	Classifier	Feature set					
			BoW	PunMM	PunMM+BoW	Syn	Syn+BoW	All(3)
Accuracy (in %)	Galaxy Zoo	DT	39.55	39.49	38.85	**39.74**	**39.55**	**39.55**
		NB	38.08	37.64	37.32	38.27	38.27	38.27
		KNN	**40.06**	**40.32**	**40.00**	39.30	39.23	39.11
	Snapshot Serengeti	DT	39.30	38.66	39.04	38.85	39.30	**40.19**
		NB	37.70	37.44	37.83	37.64	37.64	37.96
		KNN	**40.26**	**39.94**	**39.74**	**40.26**	**40.19**	39.87
F-measure (in %)	Galaxy Zoo	DT	38.79	39.17	38.25	**39.37**	**38.79**	**38.79**
		NB	37.36	36.76	34.87	37.49	37.49	37.49
		KNN	**39.85**	**40.05**	**39.68**	38.74	38.68	38.47
	Snapshot Serengeti	DT	34.42	36.89	**38.19**	35.21	34.42	30.53
		NB	37.68	**37.61**	37.61	**37.63**	**37.63**	**38.10**
		KNN	**38.08**	37.16	36.87	37.45	37.41	36.72

used in study can be put in this order based on their relative influence on the prediction of competence from text: PunMM, BoW, PunMM+BoW, Syn, Syn+BoW or BoW+PunMM+Syn. The ranking changes a bit when the Snapshot Serengeti test set is used for the evaluation i.e. BoW, Syn, Syn+ BoW or BoW+PunMM+Syn, PunM, PunM+BoW. This ranking style compares the feature sets based on their impact on a single best classifier among the three (DT, KNN and NB). There are other ways of ranking the feature sets that consider the average performance of all the three instead concerning both accuracy and F-measure.

We also tried to analyze how the Punctuation mark, the syntactic features and their combination affect of the performance of the classifiers over the Bag of Words features. Regardless the domains of the test sets involved in the evaluations, we observe that the performance of NB (BoW based) is improved by adding syntactic and punctuation marks features. Likewise, the DT (BoW based) is affected by adding syntactic and the combination of syntactic and punctuation mark features.

6 Discussion

The approaches used in this study, from user competence calculation to machine learning tasks, can be improved or possibly yield different results with alternative strategies proposed in the following paragraphs.

The most obvious approach is to use data labeled by domain experts. For Galaxy Zoo, such data is not available, but we could consider other possibilities, such as a semi-supervised bootstrapping method if we had a small amount of labeled data. Semi-supervised bootstrapping methods have been effective in various text analysis problems, such as topic and sentiment-based text classification Zhou et al. (2013). In competence estimation, to reduce dependency on majority voting, we train a classifier on a small dataset labeled by experts sampled from the training corpus. We then use the classifier to label the remaining unlabeled samples in the training corpus and retrain the classifier iteratively until we reach certain stop criteria.

Feature wise, in addition to the selected feature sets, we could use more features such as universal dependencies, character n-gram, bag-of-topics. The syntactic feature set extracted can be further enriched with features extracted using a dependency parsing to describe and represent the syntactic structure of users text better. Dependency parsing captures the dependency relationships between syntactic units/words and has been used to improve the accuracy of text classification tasks Nastase et al. (2006). As a part of improving our research results, we have also carried out preliminary experiments on a character n-gram and bag-of-topic features, where we describe a user text with topic words extracted using a topic modeling technique. We found that both types of features improve the accuracy of the trained models to a certain degree.

Finally, using multiple metadata information about users from other external data sources, for example, capturing their participations in either other seasons of the Galaxy Zoo project or other projects of Zooniverse, may help to better model the users competence.

References

D. C. Berry and D. E. Broadbent. 1984. On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, Section A. 36(2):209–231.

H. Chen and B. He. 2013. Automated essay scoring by maximizing human-machine agreement. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, 1741–1752.

Y. Chen, J. Wrenn, H. Xu, A. Spickard, R. Habermann, J. Powers, and J. D. Denny. 2014. Automated assessment of medical students? clinical exposures according to aamc geriatric competencies. *In AMIA Annual Symposium Proceedings Archive*, 375–384.

M. Dascalu, E-V. Chioasca, and S. Trausan-Matu. 2008. ASAP–an advanced system for assessing chat participants. *In AIMSA: International Conference on Artificial Intelligence: Methodology, Systems and Applications*, volume 5253 of Lecture Notes in Computer Science. Springer. 58–68.

Y. Woldemariam 2017. Predicting competence from text. *In Proceedings of The 21st World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, 147–152.

L. Fu, L. Lee, and C. Danescu-Niculescu-Mizil. 2008. When confidence and competence collide: Effects on online decision-making discussions. *In Proceedings of the 26th International Conference on World Wide Web, WWW ?17*, 1381–1390.

D. Klein and C.D. Manning. 2003. Accurate unlexicalized parsing. *In Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 423–430.

T. Kucukyilmaz, B. Cambazoglu, C. Aykanat, and F. Can. 2008. Predicting user and message attributes in computer-mediated communication. *Information Processing and Management*, 44(4):1448–1466.

D.S. McNamara, S.A. Crossley, and P. M. McCarthy. 2010. Linguistic features of writing quality. *Written Communication*, 27(1):57–86.

D. Pérez, A. M. Gliozzo, C. Strapparava, E. Alfonseca, P. Rodríguez, and B. Magnini. 2005. Automatic assessment of students? free-text answers underpinned by the combination of a bleu-inspired algorithm and latent semantic analysis. *In Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, Clearwater Beach, Florida, USA, 358–363.

M. E. Schwamb, C. J. Lintott, D. A. Fischer, M. J. Giguere, S. Lynn, A. M. Smith, J. M. Brewer, M. Parrish, K. Schawinski, and R. J. Simpson. 2012. Planet hunters: Assessing the kepler inventory of short-period planets. *The Astrophysical Journal*, 754(2):129.

V. Nastase, J. Shirabad, and M. Caropreso. 2006. Using Dependency Relations for Text Classification. *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, 12–25

Y.Haralambous, Y. Elidrissi, and P. Lenca. 2014. Arabic Language Text Classification Using Dependency Syntax-Based Feature Selection. *Proceedings of the 19th Canadian Conference on Artificial Intelligence*

G. Zhou, J. Li, D. Zhao, and Y. Feng. 2013. Semi-supervised Text Categorization by Considering Sufficiency and Diversity . *Natural Language Processing and Chinese Computing.*, 105–115.

IV

Designing a Speech Recognition-Named Entity Recognition Pipeline for Amharic within a Cross-Media Analysis Framework

Yonas Woldemariam, Adam Dahlgren

Dept. Computing Science, Umeå University, Sweden
yonasd@cs.umu.se, dali@cs.umu.se

Abstract. One of the major challenges that are inherently associated with cross-media analysis frameworks, is effectively addressing multilingual issues. As a result, many languages remain under-resourced and fail to leverage out of available media analysis solutions. Although spoken by over 22 million peoples and there is an ever-increasing amount of Amharic digital contents of various types on the web, querying them, specially audio and video contents, with a simple key-words search, is very hard as they exist in raw format. We introduce a textual and spoken content processing workflow into a cross-media analysis framework for Amharic. We design an automatic speech recongition(ASR)-named entity recognition pipeline that includes three main components: ASR, transliterator and NER. We explored and applied three different modeling techniques used for speech signal analysis, namely Gaussian Mixture Models (GMM), Deep Neural Networks (DNN) and the Subspace Gaussian Mixture Models (SGMM). The models have been evaluated with the same test set with 6203 words using the Word Error Rate (WER) metric, and obtained an accuracy of 50.88%, 38.72%, and 46.25% GMM, DNN, SGMM respectively. Also, the OpenNLP-based NER model has been developed, though trained on a very limited data. While the NER model has been trained with the transliterated form of the Amharic text, the ASR is trained with the actual Amharic script. Thus, for interfacing between ASR and NER, we implemented a simple rule-based transliteration program that converts an Amharic script to its corresponding English transliteration form.

1 Introduction

Automatic Speech Recognition (ASR) and Named Entity Recognition (NER) are commonly used Natural Language Processing (NLP) systems. They perform information extraction tasks on spoken and textual documents respectively. An ASR system generates a transcription text from speech data. ASR technologies have been used for many applications such as spoken document indexing and retrieval, speech summarization, etc, NER is used to identify and extract entity mentions, such as names of people, locations, etc from textual contents. In a text analysis task, NER serves as a pre-processing task for downstream annotators, which identifies a proper noun and classifies it into a known category. While both systems are essential to solve specific problems in isolation, they can be used together to operate on the same media, and applied in succession to add contextual information on the metadata associated with the input audio/video content for semantic search.

ASR and NER can be combined in various ways depending on the purpose of the application in question. For example, in cross-media analysis frameworks such as EUMSSI[1] (Event Understanding through Multimodal Social Stream Interpretation) and MICO[2](Media in Context), their combination is defined as an analysis workflow or analysis-chain called an ASR-NER pipeline that basically includes speech-to-text and named entity recognition services. Within these frameworks there also exist complex multimedia analysis pipelines designed to meet the requirements of complex information retrieval use cases, for instance, searching for video shots, where a person (in the shots) says something about a specific political issue using a keywords-driven approach.

Nowadays, there are plenty of multimedia extraction tools used to make searching web contents convenient. However, most of these tools are developed for well researched languages such as English and Spanish, and specific domains of applications. Due to this reason, some languages remained under-sourced including Amharic. That severely limits the access of information available in those languages. There are, however, some studies [11, 6, 8, 15] and contributions on building language technologies for Amharic, most of them are developed as proof-of-concept prototypes. As a result, it is often challenging to get computational linguistic resources for Amharic required for either NLP studies or commercial use.

Amharic is the official language of Ethiopia, spoken by over 22 million peoples, also according to the latest census carried out by Central Statistical Agency of Ethiopia[3], the second most spoken Semitic language next to Arabic. The writing system of Amharic is called “fidel”; shared with the other Semitic language of Ethiopia, Tigrinya. The basic unit has a consonant-vowel (CV) syllabic structure, usually vowels are omitted in the written form of CV, is nearly a phonetic language. There is an ever-increasing amount of Amharic digital contents of various types: text, images, audio, video, etc. on the web due to emerging information sharing platforms such as social media and video hosting sites. But searching them, especially audio and video contents, is very hard as they exist in raw format. Thus, obviously it is very demanding to have linguistically motivated multimedia analysis and extraction tools that could potentially deal with language-related concerns and make Amharic contents more searchable through keywords.

The most reasonable and affordable solution is to use open-source multilingual information extraction frameworks that provide media analysis, extraction and indexing, search and retrieval services, though they require language

[1] https://www.eumssi.eu

[2] https://www.mico-project.eu

[3] https://www.csa.gov.et

models of certain types. One of the existing open-source media analysis solutions, is the MICO platform, though it is at early stage of its release. Ideally, MICO allows extraction of multimedia contents of different languages using the corresponding language models. Within MICO, there are a number of pre-defined analysis pipelines along with their metadata extractors.

The aim of this study is to investigate adapting language specific components of MICO for Amharic. Within MICO, there are several natural language dependent multimedia analysis components such as text classification and text language detection including the ASR-NER pipeline. However, we only focus on designing of an ASR-NER pipeline for Amharic using the design principles, the standards and the technologies used in MICO. This pipeline could be considered as the first step to be able to use the MICO platform and for developing other important metadata extractors to analyze Amharic contents. Indeed, the pipeline is useful in itself, at least to index video /audio contents with extracted entities. To completely benefit from the platform more effort is needed in the direction of identifying and adapting other language dependent analysis components, for instance, sentiment analysis. We basically develop Kaldi-based ASR systems of various types, a transliterator and OpenNLP based NER extractor, to build the pipeline.

We got motivated for this study as we are one the partners of the MICO project and responsible for implementing NLP tools. While most of the implementation is done only for English, the MICO architecture allows for the integration of other language models via its API. Nevertheless, it is challenging to adapt MICO to under-resourced languages due to its requirement of trained language models that strictly satisfy the underlying design principles. This presents an opportunity to investigate the possibilities of adapting relevant language models for Amharic.

We discuss related work in Section 2, the MICO platform in Section 3, the designed ASR-NER pipeline in Section 4, the challenges and solutions in Section 5 and, finally, future works and conclusions in Section 6.

2 Related Work

There are a number of papers [3, 1] on extraction of named entities on speech transcripts on digital spoken archives for various purposes, though it is hardly possible to get any for Amharic. There are also a few research projects that investigated the introduction of an ASR-NER pipeline in multi-modal cross-media analysis frameworks for different types of languages. We primarily focus on discussing the methods used and the results achieved by these projects, as they probably best put our study into perspective, namely MICO and EUMSSI. In addition to that, although there is no published literature on the task of NER on speech transcription for Amharic, we present a brief review of research works on standalone speech recognition and named entity recognition conducted independently from each other.

During the development of the MICO metadata extractors, special attention was given to the ASR component due to the fact that most extractors, particularly text analysis components including NER heavily depend on the result produced by the ASR extractor. Thus, state-of-the-art open-source and proprietary libraries for ASR, have been well studied and evaluated against sample video contents, then the respective comparative analysis was carried out beforehand. Consequently, Kaldi[4] was chosen based on the criterion of accuracy and other technical reasons. The other good quality of Kaldi is its multi-lingual support. Most of the experiments that make use of Kaldi within MICO were effectively carried out only for English, though the MICO Showcases were planned for Arabic and Italian as well. The most challenging part of training Kaldi is that preparing a parallel corpus (speech and text) is quite costly.

Within MICO, the ASR was implemented as a speech-to-text pipeline to analyze video content and produce the corresponding text transcription in various formats. The pipeline includes audio- demultiplexing, for extracting and down-sampling the audio signal from the video, speaker diarization for segmenting information along with gender classification and speaker partitioning, speech transcription, for transcribing the audio signal into text. The resulting textual content outputted by the pipeline is further analyzed by text analysis components including the NER extractor.

The NER extractor provides a named entity extraction service on-demand when requested by other registered extractors requiring (depending) on the output produced by it. NER also takes plain text (with a text/plain MIME type) from other possible sources of textual contents such as forum discussion posts after pre-processing and parsing tasks. The NER extractor is based on the OpenNLP toolkit, that is an open-source library providing a NER service. MICO provides OpenNLP-based NER language models for English, German, Spanish and Italian, and allows an integration of models for other languages.

The ASR-NER pipeline introduced in MICO performs analysis workflows, for instance, detecting a person in a video, by collaborating with image analysis components such as the face detection extractor. Some preliminary showcases have been demonstrated by the use case partners, for instance, InsideOut10 (one of the use case partners of MICO) built a showcase application that retrieves video shots containing a specific person talking about a specific title [12].

The EUMSSI platform basically provides multimodal analytics and interpretation services for different types of data obtained from various online media sources. (their demo is available on[5]). EUMSSI seems to mainly target journalists as end users, automating their time-consuming tasks of organizing information about various events from different online and traditional data sources providing un/structured contents. The platform allows to search multimedia contents aggregated and filtered from media search engines in an interactive fashion, then enriching, contextualizing the media with extracted metadata and retrieves the result with the multimodal approach.

[4] http://kaldi-asr.org

[5] http://demo.eumssi.eu/demo/

The NER component of EUMSSI is based on Stanford NER, running on the transcription generated by ASR and text extracted by OCR from video contents, in addition to other types of textual contents from news and social media. The transcription returned from the ASR service is normalized by an auxiliary component beforehand. The ASR-NER pipeline implemented in EUMSSI, is used to annotate the speech segments uttered by each speaker shown in a video with the corresponding transcriptions and mentioned names. The resulting information is intended to get combined with the annotations obtained from the face recognition component, that enables video retrieval applications to support different search options, for instance, retrieving the quotation of peoples [10].

There are also several studies on named entity extraction on speech transcripts for independent NLP systems or audio/video analysis frameworks. For example, in the Evalita (evaluation campaign of NLP and Speech tools for Italian) 2011 workshop [4], one of the tasks was named entity recognition on transcribed broadcast news. The purpose is to investigate the impact of the transcription errors on NLP systems and explore NER approaches that cope with the peculiarities of the resulting transcripts from the ASR system.

There are a number of studies on the design and development of ASR and NER systems for Amharic. Relatively, NER is a less researched area than ASR. The survey in [11], summarizes the ASR research attempted for Amharic, ranging from syllable to sentence level detection, from speaker dependent to speaker independent speech recognition. According to the survey most of works are done using quite similar techniques i.e. HMM (Hidden Markov Model) and tools such as HTK (HMM Tool Kit). There is an attempt to develop and integrate an ASR system into the Microsoft Word application to enable it to receive file related commands. The survey also pointed out that the major reasons, why the ASR systems failed to be used in speech applications, to mentions some of them: they are trained on read speech with a limited dataset and fail to handle germination and morphological variation. There are also a few unpublished research works on Amharic NER [9, 8]. The recent work [2] introduced deep learning with the skip-gram word-embedding technique by extending the previous works. The authors in [2] developed Amharic NER prototypes using the same method i.e., Conditional Random Fields and the same corpus as in [9, 8] but different subsets, and obtained different results.

3 The MICO Platform

MICO is a cross-media analysis plaform, which provides media analysis, metadata publishing, search and recommendation services (described in [14]). Generally, MICO has three types of metadata extractors, textual extractors for performing linguistic analysis such as parsing, sentiment analysis, text and classification, image extractors for performing image analysis for detecting and human faces and animals from images, audio extractors for performing different speech analysis tasks such as detecting whether audio signals contain music or speech, and extracting audio tracks from video content and producing a transcription.

Within the MICO platform metadata extractors interact and collaborate each other in automatic fashion via a service orchestration component (aka broker) to put a media resource in context. Several semantic web technologies such as Apache Marmotta[6] and SPARQL-MM[7] have been used for storing the metadata annotation of analysis results in RDF format and querying the metadata respectively. The Apache Hadoop[8] distributed file system used for binary data, and Apache Solr[9] for the full-text search.

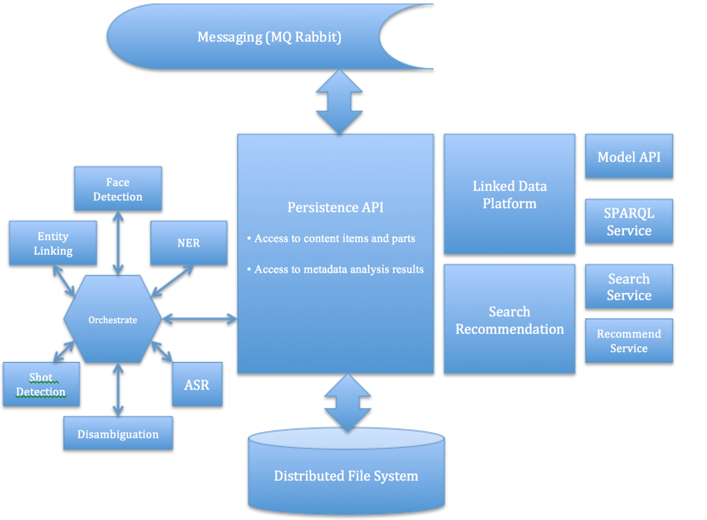


Fig. 1: The Architectural Structure of the MICO Platform

4 The ASR-NER Pipeline

The Amharic ASR-NER pipeline designed for this study includes three main components: ASR, transliterator and NER (see Figure 2). The pipeline performs extracting named mentioned from the input audio/video data. Within the MICO architecture, the core ASR component needs to be connected with the pre-processing and post-processing components, that forms a speech-to-text sub-pipeline.

There are two pre-processing components, namely audi-demux and LIUM diarization. The former does extracting audio tracks from a video input and down-sampling the audio tracks. The later does segmenting the audio tracks

[6] http://marmotta.apache.org
[7] http://marmotta.apache.org/kiwi/sparql-mm.html
[8] http://hadoop.apache.org
[9] http://lucene.apache.org/solr/

into smaller units using gender and speaker information. The post-processing component, namely XML2text transforms the output file in text/xml format generated by the core ASR component to plain text (text/plain) required by the NER component.

4.1 The Implementation of the Amharic ASR

We explored and applied three different modeling techniques, namely Gaussian Mixture Models (GMM), Deep Neural Networks (DNN) and the Subspace Gaussian Mixture Models (SGMM) to implement Kaldi-based Amharic ASR systems. As a result, three acoustic models have been generated using each technique with the parallel speech-transcription corpus and the pronunciation lexicon provided on the ALFFA[10] project. We also used the 5-gram language model described in [7]. Originally, the raw corpus was prepared for the study in [5], it is about 20 and 2 hours of speech for training and decoding respectively.

All the three models have been trained using 13 Mel-frequency cepstrum coefficients (MFCCs) features, followed by linear discriminant analysis (LDA) and transformation, maximum likelihood transform (MLLT). Also, Feature-space maximum likelihood linear regression (fMLLR) has been used as speaker adaptation technique. The models have been evaluated with the same test set with 6203 words using the Word Error Rate (WER) metric, and obtained an accuracy of 50.88%, 38.72%, and 46.25% GMM, DNN, SGMM respectively. Compared with state-of-the art ASR systems built for Engslish, for instance, authors in [13] achieved a 5.1% WER, more tasks are needed to improve our ASR.

4.2 The Amhairc NER model and the Transliterator

We use the OpenNLP-based NER model developed for Amharic by Mikiays [8]. As the original model was prepared using the format used in [8], the data needs to be re-labeled manually to train the OpenNLP model, that is the format supported by MICO. However, it was possible only to label the small portion of the data used in [8] and limited to identify persons, locations and organizations. The model is trained using the algorithm provided by openNLP: MaxEntropy, Perceptron, and Naive Bayes.

While the NER model has been trained with the transliterated form of the Amharic text, the ASR is trained with the actual Amharic script. Thus, to support the interfacing of ASR with NER, we implemented a simple rule-based transliteration program that converts an Amharic script to its corresponding English transliteration form.

5 Challenges and Solutions

Since the main goal of this research is to make under-resourced languages beneficial out of the media analysis technology built for resource rich languages,

[10] https://github.com/besacier

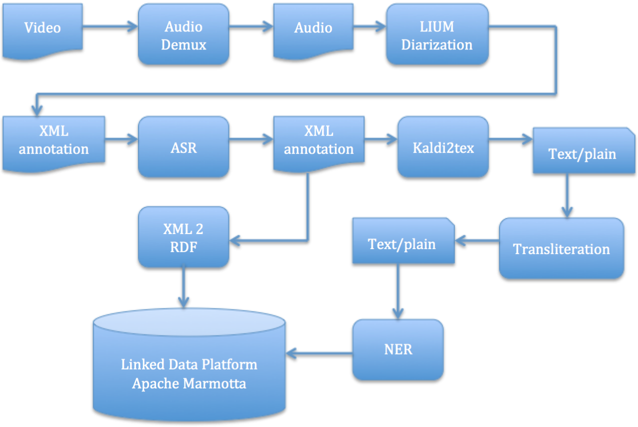


Fig. 2: An ASR-NER Pipeline within a Cross-media Analysis Framework

by dealing with the issues related with scarcity of computational linguistic resources, most of the challenges faced in the course of the research is inherently associated with the lack of resources. In addition, we assumed that the resources that have been available can be modified with reasonable amount of configuration tasks and then would fit to the designed experimental settings, but a number of evaluations (compatibility tests) have shown that they turned out to require to get transformed with much amount of works. For example, re-labeling the NER dataset, converting the language model to the finite-state-transducer format and so on.

The other problem regard related computational resources, training the DNN model has been challenging due to the requirement of a GPU processor along with the queue scheduling service configuration. Although it is extremely slow, the training has been done on our CPU machine with a slight job-scheduling configuration task.

Lastly, it concerns the interfacing of ASR with NER. The transcription generated by ASR is in actual Amharic script, where as the NER model is trained on the English-transliteration form of the Amharic text. Thus, to support NER a simple rule-based transliteration program has been written.

6 Conclusion and Future Work

We identified language dependent analysis components that are viewed as a high priority including ASR and NER, within a cross-media analysis platform. We designed an ASR-NER analysis pipeline for Amharic based on state-of-the art design principles and techniques employed in cross-media solutions. That promotes the multi-language support of the MICO platform. Generally, other languages somehow take advantages of the methods proposed here, especially they can be easily extended for Semitic ones such as Tigrigna.

Now, other language-oriented and audio-visual extractors can build on the top of the pipeline to completely exploit the automatic-annotation utilities of the MICO platform. However, that requires to go through all the extractors and discover the synergy between them, which might include developing auxiliary components to support dynamic interactions during real-time executions.

We found that the DNN model outperforms than the GMM and the SGMM models. However, as the NER models are trained on a very small dataset we could not run standard pipeline evaluations. Thus, we need to improve the performance of the NER model with large corpus to measure the overall quality of the pipeline. We are also interested to improve the accuracy of the DNN model using lower gram language models such as trigrams, as well as using high performance computational resources such as a GPU installed machine, which allow us to run several tests without waiting for a long time to fine-tune models parameters and optimize the ASR system.

7 Acknowledgment

We acknowledge the financial support from the EU FP7 MICO project. We also thank Mikyas Belay for providing the NER model used in this study.

References

[1] Hori A and Atsushi N. An extremely large vocabulary approach to named entity extraction from speech. In *IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, pages 973–976, 2006.

[2] G. Björn and S. Utpal. Named entity recognition for amharic using deep learning. *2017 IST-Africa Week Conference (IST-Africa)*, pages 1–8, 2017.

[3] M.F.M Chowdhury. A simple yet effective approach for named entity recognition from transcribed broadcast news. In *Evaluation of Natural Language and Speech Tools for Italian*, pages 98–106, 2013.

[4] B. Magnini, F. Cutugno, M. Falcone, and E. Pianta. *Evaluation of Natural Language and Speech Tools for Italian. Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg, 2013.

[5] T. Martha, A. Solomon, and B. Laurent. Using different acoustic, lexical and language modeling units for asr of an under-resourced language - amharic. *Speech Communication*, 56, 2014.

[6] Y. Martha. Automatic amharic speech recognition system to command and control computers. Master's thesis, School of Information Studies for Africa, Addis Ababa University, 2003.

[7] M. Michael, B. Laurent, and M. Million. Amharic-english speech translation in tourism domain. In *SCNLP@EMNLP 2017*, 2017.

[8] B. Mikiya. Amharic Named Entity Recognition Using a Hybrid Approach. Master's thesis, School of Information Informatics, Addis Ababa University, 2014.

[9] M. Moges. Amharic Named Entity Recognition. Master's thesis, College of Natural Sciences, Addis Ababa University, 2010.

[10] L. et al Nam. Towards large scale multimedia indexing: A case study on person discovery in broadcast news. In *In Proc. of International Workshop on Content-Based Multimedia Retrieval*, 2017.

[11] A. Solomon, T. Martha, and M. Wolfgang. Amharic speech recognition: Past, present and future. In *In: Proceedings of the 16th International Conference of Ethiopian Studies*, pages 1391–1401, 2009.

[12] K. Thomas, S. Kai, and K. Harald. Enabling access to linked media with sparql-mm. In *WWW*, 2015.

[13] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke. The Microsoft 2017 Conversational Speech Recognition System. *ArXiv e-prints*, 2017.

[14] W. Yonas. Sentiment analysis in a cross-media analysis framework. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pages 1–5, 2016.

[15] W Yonas and H. Sebsibe. Duration modeling of phonemes for amharic text to speech system. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 1–7, 2012.