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

1 Introduction **1**

2 Conceptual Backgrounds on NLP **3**
2.1 Text Analysis 3
2.1.1 Sentiment Analysis 5
2.1.2 Competence Analysis 5
2.2 Machine-Learning Methods in NLP 6
2.2.1 Naive Bayes 6
2.2.2 Decision trees 7
2.2.3 K-Nearest Neighbor 8

3 NLP Tasks in Cross-Media Analysis Frameworks **9**
3.1 The MICO platform 10
3.2 Automatic Speech Recognition (ASR) 10
3.3 Named Entity Recognition and Linking 12

4 Summary of Contributions **15**
4.1 Sentiment and Competence Analysis 15
4.1.1 Paper I: *Sentiment Analysis in a Cross-media Analysis Framework.* 15
4.1.2 Paper II: *Predicting User Competence from Text* Paper III: *Predicting User Competence using Linguistic Data* . 16
4.2 Speech and Named Entity Recognition 17
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* 17

5 Future Work **19**

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:

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

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

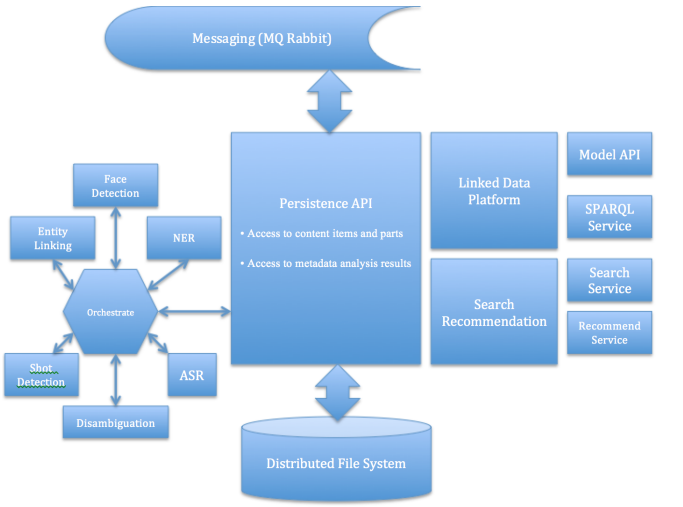

Figure 3.1: The Architecture of the MICO Platform, adopted from [13]

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

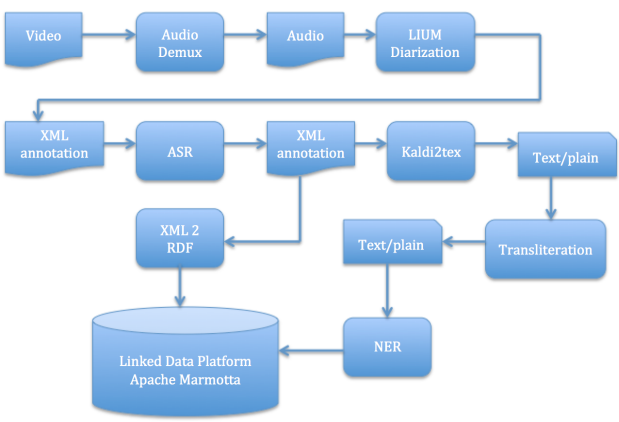

Figure 3.2: An ASR-NER Pipeline within a Cross-media Analysis Framework

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, fol-

lowed 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.