Extracting Attributes for Online Communities

A thesis submitted in partial fulfilment of the requirement for the degree of Doctor of Philosophy

Ashwaq Mohammed Alsulami

School of Computer Science & Informatics
Cardiff University

July 2023

Abstract

Numerous organisations frequently require insights into social media discussions, including identifying trending topics and understanding the characteristics of individuals participating in these discussions. Numerous methods have been suggested to extract attributes that can effectively characterise a group engaged in a conversation. Some of these methods rely on supervised learning, which requires a substantial volume of labelled data. Others are bespoke techniques, which can only be applied to certain attributes, for example, using language models to detect that a tweet is written by a person of what age. These methods lack scalability to capture a broader range of attributes because they either require a prohibitively expensive process for data labelling or can only deal with some specific attributes.

In this thesis, we propose an unsupervised learning approach to extracting attributes from user profiles, aiming to address the scalability issue associated with the existing methods. Our approach consists of two stages. In the first stage, lexical sources and semantic analysis are used to determine whether a user in their profile description suggests a particular attribute. In the second stage, we use the results from the first stage as training data to train a classification model to determine the attribute for users whose attribute cannot be identified in the first stage.

Our findings demonstrate that our approach to detecting attributes in discussion groups can capture attribute from user profiles without the need for data labelling. We have effectively implemented our methodology across a set of attributes, obtaining an average accuracy of 78% in attribute extraction. We have effectively examined the application of the developed method and determined the percentage of users within a given hashtag community exhibiting a specific attribute. This analysis has provided valuable insights into the characteristics of the group.

Contents

List of Publications

- Alsulami, A. and Shao, J., 2022. Extracting Attributes for Twitter Hashtag Communities. International Journal of Humanities and Social Sciences, 16(3), pp.171-178 [3].

List of Figures

List of Tables

List of Algorithms

Acknowledgements

I would like to express my grateful to my supervisor, Dr. Jianhua Shao, for his invaluable guidance and unwavering support throughout my PhD study. I would also like to extend my heartfelt thanks to my family and friends for their continuous support and encouragement, which have been instrumental in my success.

Chapter 1

Introduction

Nowadays, it is very easy for people to connect and communicate with each other virtually or online due to the availability of modern social network platforms. This has resulted in a large amount of user interaction data that can be analysed to support a range of applications. One such application is the discovery of online communities on social networks, in which people attempt to find groups of users who, for example, share the same interests [46, 47], are connected in a certain manner [8], communicate regularly with each other [4] or hold the same opinions on specific topics [11]. The growth of online communities across various platforms has generated an interest in understanding their demogrphics.

Determining a community's demographics, such as gender, age, and race, is useful. This is because it can be used to support, for example, personalised advertising or enhance content recommendation. For example, assume that there is a Twitter community (groups of users who discuss the same topic) for a certain type of sporting event. If we found that the majority of this group comprised women between the ages of 20 and 30, then a sportswear company might advertise products that are suitable for women in that age group to this community.

In recent years, numerous methods have been proposed for extracting attributes from social media users. However, some of these studies rely heavily on supervised learning, which is trained on a substantial volume of labelled data, while others develop bespoke methods that work only with specific types of data. For example, language models can be utilised to determine the age group of a user based on their tweets and image processing can be employed to detect a user's age from pictures. Our work uses semantic analysis as technique to classify users' profiles, making it more scalable and applicable. In this study, we attempt to extract attributes from communities without relying on labelled data.

1.1 Extracting User Attributes from Communities

To derive an understanding of the characteristics of a community is challenging, because people may not always share their demographic information explicitly on social platforms. A number of methods [91, 83, 101, 85, 34, 98] have been proposed to extract user demographics from social media users through various inferences. Some methods have used content (such as posts), some have used social connections (such as friendship and follower relationships) and others have used profile information (such as user's photo, first name and description). For example, Schwartz et al. [83] estimated the age of Facebook users based on the words used in their postings, and Vijayaraghavan [91] analysed users' first names to infer their gender, picture features to extract their age and gender, and information regarding followers or who they follow to predict their political orientation and location.

However, these methods have two main weaknesses. First, the majority of them are based on supervised learning, which requires labelled data for training [12]. However, data labelling has limitations, including quality, cost and slowness of the process as well as a lack of ground truth. If we wish to develop a general method that is capable of extracting a large range of attributes from a community, then labelling the data would be very costly, as training data for each such attribute would need to be labelled. For example, labelling profiles with genders is necessary if we want to identify a user's gender. Similarly, if we want to identify a user's level of education, we must label the data with education information.

Second, previous studies have employed bespoke techniques designed for specific attributes. For example, Sloan et al. [85] inferred age and occupation from profile descriptions, and Messias et al. [58] extracted gender and race attributes from pictures. Their methods can extract certain attributes, but are only effective for those attributes and completely useless when used to obtain other attributes because of the techniques they employed, such as using language models and image processing. As such, their methods lack scalability.

1.2 Proposed Approach

Our approach attempts to address the weaknesses of previous research on attribute extraction. It is unsupervised, which eliminates the need for labelled data. We employ semantic analysis to determine whether the user's profile contains specific attribute, enhancing scalability and applicability to any attribute of interest.

We apply this approach to Twitter #hashtag communities, which are groups of Twitter users who tweet on the same #hashtag by analysing the profile data of the participants. This study proposes a two-stage methodology involving semantic analysis and machine learning to extract attributes from user profiles within communities. This approach extracts any attribute of interest without relying on human labelling.

The proposed approach uses semantic analysis to overcome the limitations of existing research. Our approach consists of the following two main stages.

1. Lexicon-based attribute extraction (LBAE) stage, which uses a lexical source and some semantic analysis to determine if a profile has a certain attribute.
2. Classification enhanced attribute extraction (CEAE) stage, which uses the result of the LBAE stage as ground truth to train a classification model to determine the profiles that cannot be determined in the LBAE stage.

Consider the community given in Table 1.1, for example. Suppose that we wish to determine if this community has `Christianity` as faith. The LBAE stage will first expand `Christianity` using a lexicon source, such as WordNet, to find its synonyms, as presented in Table 1.2. We then calculate the semantic similarity between these values and words that occur in a profile. A profile is considered to have the `Christianity` attribute if the semantic similarity score is sufficiently high. In our example, `Catholic`, `Jesus` and `Christian` all have sufficient similarity to the synonyms `Catholicism` and `Christianity`, as shown in Table 1.1; therefore, u_1, u_2, u_3 and u_4 are all considered to have the `Christianity` attribute. However, u_5, u_6 and u_7 profiles are not categorised at this stage because they lack sufficient similarity to synonyms. The CEAE stage builds a model utilising the user profiles of u_1, u_2, u_3 and u_4 that have been classified. When the model is used to categorise the remaining profiles (u_5, u_6 and u_7), u_5 and u_6 are classified to have a `Christianity` faith (due to the term `church` in the classification model), which leaves only u_7, whose faith is still unknown. After considering the two stages, the profile of u_7 does not fit within the classification of `Christianity`. Upon examination of the profile, it appears that it should not be categorised as such. As evident from the example, in contrast to existing work, our approach to attribute extraction is unsupervised and does not require labelling the data. As such, it can improve the scalability and generality of attribution extraction for communities.

Table 1.1: An example of the #hashtag community

User (U)	Profile (P)	LBAE stage	CEAE stage
u_1	prolifer Catholic Love Jesus proud mom of Iraq war veteran avid supporter of our military avid animal lover addicted to reruns of Monk he is so funny Mr. Monk.	Christianity	Training data
u_2	Ordained Elder In the Church of Jesus Christ. Ordinances There is No Other Way!	Christianity	
u_3	MATT 11.27 the Lord willed to reveal Himself through me to the Baptist Church in 2004. A man of Faith in God and believing that Jesus Christ is His Son, Faith	Christianity	
u_4	Prostate cancer survivor; husband, swimming daddy, Christian daddy, leadership survivor, church survivor, grateful for Aba father, love Jesus andThe Holy Spirit	Christianity	
u_5	I have the unmitigated gall to actually live what I believe.Believe in church planting. Biblicist. #TGDN freedom, liberty, salvation retweets not endorsement.	Unknown	Christianity
u_6	Senior pastor /president of City voice of fire gospel int'l church worldwide, songwriter, psalmist and Blessed with Apostolic and prophetic Grace. #Bible	Unknown	Christianity
u_7	Mother of 6 who I love more than life. Simple I am, and ladylike I try to be. My favorite colors are pink and navy, and pearls are a must. #kindnesswins #maga	Unknown	Unknown

Table 1.2: Expanded words for Christianity using WordNet

Attribute	Expanded words
Christianity	Christianity, Adventism, Albigensianism, Catholicism, Donatism, Protestantism, Tractarianism, Puseyism. . . . etc.

1.3 Research Contributions

Our observation is that supervised learning does not offer a sufficiently scalable approach to attribute extraction because we often do not know which attribute a user wishes to extract from a community. Even if we had this information, labelling data for each attribute would be rather costly.

The study presented in this thesis adds the following to the body of literature on attribute extraction:

1. It demonstrates how attributes can be found using semantic similarity.
2. It establishes a framework for automatically labelling user profiles by utilising semantic similarities and subsequently employs the obtained results to train supervised learning algorithms.
3. It offers the first generic approach to detecting any attribute desired by the user.

The main contribution of this thesis is the use of unsupervised learning in attribute extraction. The incorporation of unsupervised learning into our approach enables its applicability to any attributes without labelled data. More specifically, this study makes the following contributions.

- We propose a methodology that aims to extract attributes from profiles of community participants. This methodology consists of two stages: the LBAE and CEAE stages. The LBAE stage deals with analysing the semantic relationship and similarity between the interested attribute and a profile to determine if an attribute is present. Then, the results of the LBAE stage are subsequently used to build the classification model in the CEAE stage. The main objective of this methodology is to eliminate the need to label the initial dataset by employing semantic relationship and similarity analysis.

- We propose the LBAE stage to extract attributes from a given profile. Our method attempts to identify the attributes of community members and determine the proportion of users possessing these attributes. The objective of this stage is to extract attributes from user profiles without labelling the data. This is achieved by using lexicon resources such as WordNet and semantic analysis. In contrast to other studies, we rely on an unsupervised learning method that is more scalable and can be used to work with any given attribute.

- We propose the CEAE stage that enhances the result from the LBAE stage. While the LBAE stage can identify a set of people who have a particular attribute, certain profiles cannot be classified. This stage attempts to categorise people who cannot be categorised by the first stage. The CEAE stage builds a classification model learned from the LBAE stage results. The objective of this stage is to classify more profiles. This is achieved through iterative learning, which involves repeated training processes.

- We test our approach using hashtag communities with different properties to ensure that our research is conducted in a variety of situations. We choose topics from both general and specialised areas. Our experiments reveal that our methodology can be applied to many attributes, without needing human intervention to label any data.

1.4 Research Aims and objectives

This research aims to analyse community participant profiles to determine the proportion of users possessing a specific desired attribute. We developed an unsupervised methodology to extract the desired attributes from community members. It seeks to address the question of how effectively semantic analysis can extract attributes without relying on human labelling. The challenge of acquiring labelling data for every desired attribute limits the applicability of supervised learning models to only attributes with available labelling data.

The objectives of this research are as follows:

1. Develop a semantic analysis based technique to extract attributes without labelling the data.
2. Use a classification learning model to identify the attributes of additional community members and determine the proportion of users possessing these attributes. This involves a continuous learning process, where training is repeated to classify more community members.
3. Test the effectiveness of the extraction methodology through case studies on various online communities.

1.5 Thesis Structure

The remainder of this thesis is organised in the following manner:

In Chapter two, we discuss the concept of community in general. We survey existing work in the field of attribute extraction from social media users. We end the chapter by reviewing the research on establishing semantic relationships.

In Chapter three, we define the attribute extraction problem and discuss the two main approaches for resolving it: bottom-up derivation and top-down derivation. We then provide an overview of our two-stage approach—the LBAE and CEAE stages—for extracting attributes from communities. We then describe the LBAE stage in detail. This work has been published in [3]

In Chapter four, we explain how the CEAE stage is used to classify more community members whose attribute cannot be classified using the LBAE stage. In addition, we discuss a problem that the CEAE stage encounters while classifying attributes and its potential solutions.

In Chapter five, we report experiment results. We describe the datasets used in our study and the data pre-process. Thereafter, we evaluate the the effectiveness of the proposed approach

In Chapter six, we conclude the thesis and summarise our findings. Furthermore, we discuss prospective future work.

Chapter 2

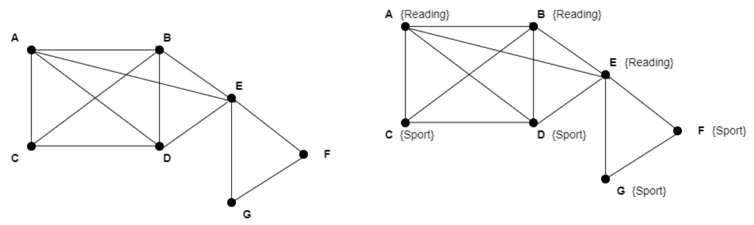
Background and Literature Review

In this chapter, we discuss the literature that is relevant to our study in this thesis. Our research focuses on attribute extraction from on-line communities (communities for short from now on). We explore relevant literature on attribute extraction.

The remainder of this chapter is organized in the following manner. We review the idea of online community in Section 2.1. Then, we examine what has been accomplished in the field of attribute extraction for online communities in Section 2.2. In Section 2.3, we discuss related work on semantic relationships and similarity, as our work relies on the assessment of semantic similarities in identifying attributes.

2.1 Community Concept

A social network can be abstractly represented as a graph, where vertices or nodes represent the individual participants in the social network, and edges or links represent the relationships among them. Figure 2.1 illustrates an example of a social network represented as a graph and an attributed graph. An attributed graph is a type of graph in which each vertex has one or more attributes associated with it. Figure 2.1b depicts an example of an attributed graph, in which each vertex represents a user in a social network, and is associated with keywords that describe the user's interests. In numerous studies and applications, a community is defined as a group of nodes or vertices within a graph, that are connected in a certain manner.



(a) Social network as graph

(b) Social network as an attributed graph

Figure 2.1: Representation of social network

An online community is a group of people who connect, share certain common interests and communicate with each other using the internet. For example, a community may be formed around a shared interest, such as a sports team, book club, or gaming group. Many social network communities have been defined in previous research. For example, groups of users who have the same interests can be considered as forming communities [46, 47]. In [8], Bakillah et al. considered communities as people who were situated close to each other (geo-locate). Another study defined communities as groups of users who regularly communicate with each other [4]. Cao et al. [11] defined communities as users with the same opinions on specific topics.

In this thesis, we consider a group of Twitter users who tweet on the same #hashtag (a word or phrase preceded by a hash sign #) discussion as a community, and our aim is to extract attributes that characterise the participants in this community.

2.2 Attribute Extraction

Many studies have attempted to automatically infer demographic attributes for communities. In this section, we review methods for attribute extraction. First, we discuss the types of data that have been used in attribute extraction. Next, we illustrate which attributes have been considered in previous studies. Finally, we review the techniques used to extract these attributes.

2.2.1 Data Used for Attribute Extraction

Studies on attribute extraction can be broadly categorized into three groups based on the type of data used for extraction: image-based attribute extraction, network-based attribute extraction and text-based attribute extraction. In this section, we provide a detailed description of each of these categories.

Image-based Attribute Extraction

Several studies [91, 13, 58, 35, 5, 102] have attempted to infer user demographics through image processing of profile pictures and other multimedia such as images and video posts. For example, Vijayaraghavan et al. [91] used profile images as input to a deep learning model that can classify Twitter users' demographics. Chakraborty et al. [13] used images of users who contributed to trend topics to infer their demographics. Messias et al. [58] attempted to identify the gender and race of Twitter users in the United States using advanced image processing algorithms from Face++. Huang et al. [35] conducted an analysis on communities associated with hate speech, where they extracted age, gender and race/ethnicity from users' profile images. An and Weber [5] analyzes the demographics of Twitter users who use certain hashtags, such as #greysanatomy and #yankees. This study examines the relationship between demographics and hashtag use, including gender, age, and race. Zagheni et al. [102] utilised facial recognition software (Face++) to estimate a user's age within a 10-year range.

These studies demonstrate the potential of using images to infer the demographic attributes of Twitter users. Image-based attribute extraction methods have limitations, such as accuracy being influenced by image quality, requiring a lot of labelled training data and being computationally expensive. It is also restricted to visual attributes, such as age, gender and emotion, and cannot extract non-visual attributes like religion, job, and hobbies. Using images alone may not be sufficient to accurately infer demographics because users may use images that do not reflect their actual demographics. An important limitation of this approach is that age estimates may be biased. For instance, users may have posted pictures of themselves when they were younger or may not have updated their profile pictures. Therefore, it is important to consider multiple modalities, such as text and network information, when inferring demographics. In comparison, our study focuses on extracting the attributes of communities based on text.

Network-based Attribute Extraction

Some previous studies [15, 91, 71] have inferred users' attributes based on networking, such as their followers, friends and retweets. For example, Culotta et al. [15] proposed a method to extract Twitter users' gender, age, ethnicity, education, income and child status based on website traffic data. They utilised a tool called "Quantcast" to collect website traffic data from a selected group of Twitter users. This tool provided information such as age and gender of website visitors. Thereafter, they cross-referenced this website traffic data with the respective Twitter user profiles to extract demographic attributes. The study only includes Twitter users who visit websites that use the Quantcast tool, which could potentially introduce biases and limit the generalisability of the results to the broader Twitter population. Vijayaraghavan et al. [91] presented a demographic classifier for gender, age, political orientation, and location on Twitter. They use information regarding followers or who they follow to predict their political orientation and location. Other attributes were predicted using image-based or text-based methods. This study collected Twitter demographic dataset for this task using a deep multimodal multi-task learning architecture. Pan et al. [71] also used network-based approaches to predict the occupation of Twitter users.

These studies use different approaches to predict the demographics of users. Culotta et al. [15] used regression to predict six demographic variables of a set of Twitter users based solely on whom they follow. Vijayaraghavan et al. [91] presented a demographic classifier for gender, age, political orientation, and location on Twitter using a deep multimodal multi-task learning architecture. Pan et al. [71] used homophily, the tendency of individuals to associate with others who are similar to themselves, to predict the occupation of Twitter users based on the network of Twitter users and their followers. These studies focused on predicting the demographics of users using network-based approaches. In contrast, our study uses a text-based approach.

Network-based attribute extraction methods have limitations in inferring attributes solely from users who follow specific popular accounts. Additionally, these techniques fail to account for the scenarios in which social network users follow celebrities with multiple attribute values, such as different occupations.

Text-based Attribute Extraction

Other previous studies [83, 85, 101, 98, 41] have extracted attributes based on textual features, such as content of posts, location and self-description. For example, Schwartz et al. [83] explored how personality, gender and age can be inferred from language used in Facebook. They detected language features (words, phrases and topics) of many Facebook posts via open-vocabulary analysis, which enables analysis of all words used by individuals on social media, rather than merely a pre-defined set. They found that certain words and language patterns are correlated with specific personality traits, genders, and age groups. For example, females tend to use words related to positive emotions and social behavior, while males use more words related to anger and swearing. Although the open-vocabulary approach enables an examination of language use, it results in numerous statistically significant associations that lack practical significance or value in predicting personality, gender, or age. Another limitation is that their model was only able to identify the age and gender of users, without being able to capture other demographic attributes. Furthermore, the generalisability of the study's findings to other cultures or populations beyond Facebook users primarily from the United States is limited, as the study primarily focused on a sample of Facebook users from the United States. Sloan et al. [85] conducted a study to determine the age, occupation, and social class of Twitter users by analysing their profile descriptions. They utilised pattern-matching to find the attributes. A limitation of their method is its reliance on data matching. Analysing additional attributes would require separate data matching techniques for each attribute of interest. Yo and Sasahara [101] used machine learning techniques to infer personal attributes such as age, gender, and occupation from Twitter user metadata. Wood-Doughty et al. [98] proposed a neural model that infers age, gender and ethnicity from a user's name and screen name. They found that their approach was able to accurately predict gender and age with 85% and 72% accuracy, respectively. However, accurately predicting ethnicity was challenging, thereby indicating that the approach may not be effective for all demographic categories. This approach requires a large labelled dataset (over 20 million Twitter users) to train the machine learning models, which can be time-consuming and costly. Klein et al. [41] used self-reported age information in tweets to automatically extract the exact age of Twitter users.

These studies used machine learning techniques and self-reported information to infer demographic characteristics. The studies that used machine learning techniques achieved higher accuracy than those that used self-reported information. These studies showed that text-based analysis of social media data can be used to infer various attributes. However, our proposed approach does not require labelled data and relies on lexical sources to detect attributes for Twitter discussion groups.

Text-based attribute extraction methods have several advantages, including a wider range of attributes such as religious beliefs, that can be extracted. Therefore, in this thesis, we consider text-based attribute extraction. More precisely, we utilise user profiles to extract attribute. Using user profiles can be less intrusive for users, as user profiles only collect information that the user is willing to share. A few studies [65, 63] have shown that user profiles are useful for predicting user demographics.

2.2.2 Attributes that have been Considered

Previous studies have concentrated on identifying a few specific user attributes. For example, Hu et al. [34] proposed a method that extracts occupation of the participants based on tweets, Sloan et al. [85] inferred age and occupation, Messias et al. [58] extracted gender and race, and Vijayaraghavan et al. [91] analysed users' first names to infer their gender, picture features to extract their age and gender and information regarding followers or who they follow to predict their political orientation and location. Table 2.1 displays the attributes considered in the previous works.

Previous studies have tended to focus on finding certain specific attributes only. While these studies can effectively extract these attributes, they are limited to those specific attributes and cannot be used to obtain other attributes. For example, Messias et al. [58] used advanced image processing algorithms from Face++ to determine the gender and race of Twitter users in the United States. However, determining other attributes from images poses challenges, necessitating supplementary data sources or alternative methodologies. The image processing algorithm is constrained to analysing visual characteristics such as age, gender and emotions and is thus unable to extract non-visual attributes such as religion, occupation and hobbies. In this study, we propose an unsupervised method that can be used to extract any attribute.

2.2.3 Techniques for Attribute Extraction

In the previous sections, we discussed previous research on attribute extraction in terms of the source of data used and the types of the attributes that can be extracted. In this section, we review the techniques employed to extract attributes, which can be categorised into two broader categories: rule-based and machine learning-based.

Rule-based Techniques

Rule-based techniques involve creating a set of predefined rules or patterns to identify attributes within the data. These rules may be based on regular expressions, or other forms of pattern-matching techniques. For example, Huang et al. [35] inferred geographic location by matching regular expressions. Al Zamal et al. [2] and Morgan-Lopez et al. [67] identified users' age by searching for self-reported birthday announcements in tweets (e.g., Happy 39th birthday to me). Both approaches are straightforward and efficient, offering the advantage of inferring the age of users who have not explicitly disclosed this information. However, their applicability is constrained to users who have shared their birthday information.

Mislove et al. [64] used a gender inference method that involved matching first names of US-based Twitter users to census data. This method relied on analysing the frequency of certain names and the associations between them and genders in a given culture or society to infer users' gender. This method is also simple and effective; however, some first names are more commonly associated with either men or women, but not exclusively. Furthermore, individuals may opt for gender-neutral names or nicknames, which pose challenges in accurately inferring their gender.

Wood-Doughty et al. [99] provided a technique for extracting self-reports of race and ethnicity from Twitter profile descriptions. They employed a keyword-matching approach to identify a substantial corpus of Twitter users who indicate self-identification with a specific racial or ethnic group. The four main racial or ethnic groupings (White, Black, Asian, and Hispanic/Latinx) in the US are the only ones taken into account by this model; smaller populations and multiracial categories are ignored.

Although these methods have provided valuable insights into the extraction of attributes through rule-based techniques, it is crucial to acknowledge their limitations, as the accuracy of the results may vary based on the dataset and the attributes under consideration. For instance, Huang et al. [35] achieved an accuracy of 83% in inferring geographic location from English tweets, but this accuracy decrease when applied to tweets in other languages. Similarly, Al Zamal et al. [2] reported a 90% accuracy in inferring age for users who self-reported their birthday on Twitter, whereas Morgan-Lopez et al. [67] achieved an 88% accuracy for the same attribute. Our methodology's first stage shares with previous studies by employing rule-based techniques for attribute extraction. Nevertheless, our approach differs significantly. First, our study automates the generation of rules and evaluates their similarity to user profiles to determine attribute presence using lexical sources such as WordNet. In contrast, previous studies required authors to manually specify rules for each attribute. Second, our study can be applied to any attribute, unlike previous studies that are tailored to specific attributes. For example, Al Zamal et al. [2] and Morgan-Lopez et al. [67] focused on age inference, Mislove et al. [64] focused on gender, and Wood-Doughty et al. [99] focused on race and ethnicity.

One advantage of using the rule-based techniques is their reliance on simple and often easy to capture user information, thereby enabling researchers to categorise users with a high degree of certainty. However, each attribute requires a different set of pattern-matching techniques to drive it based on rules. Therefore, if we were to develop a general method that can extract any attribute from a community, the cost of implementing a set of data matching techniques for each attribute would be even more costly.

Our study contributes to the field by demonstrating the capability of automatically generating rules for attribute extraction. While rule-based techniques prove to be valuable tools in attribute extraction, it is imperative to complement them with other methodologies, notably machine learning. Machine learning excels in capturing patterns related to attributes that may not be explicitly defined by the rules. Therefore, employing a combination of rule-based techniques and machine learning enhances the efficacy of attribute extraction methodologies.

Machine Learning-based Techniques

Techniques based on machine learning involve collecting data from various sources, such as text, images or networks, and using that data to build models capable of predicting attributes. For example, Pennacchiotti and Popescu [76] propose a method to classify Twitter users based on their tweets to extract political affiliation and ethnicity. The method involves extracting various features from Twitter data, including tweet content, and sentiment analysis. These features are then used to train several machine learning models to classify the users. Ludu [54] proposed a machine learning-based approach to predicting the gender of Twitter users based on the gender of the celebrities they follow. To create a training dataset, data on the genders of celebrities were collected and utilised to train models. The trained models were then used to predict the gender of Twitter users based on the gender of the celebrities they follow. Vicente et al. [89] predicted the gender of Twitter users based on multiple information, such as user name and screen name, user description, content of the tweets, and profile picture. They used a dataset of over four million Twitter users and achieved an accuracy of over 85% in predicting the gender of these users. Volkova et al. [92] used natural language processing techniques to extract users' interests by analysing the language and posting behavior of Twitter users. This involved identifying topics discussed and hashtags used in tweets as well as analysing posting frequency, number of followers, and time of day active on the platform. Machine learning algorithms were then used to infer users' age, gender, and personality traits based on these extracted features. For example, certain language features like emoticons and exclamation marks were linked to younger users. These methods rely on a labelled dataset to learn the relationship between features and the target attribute. Therefore, to generalize the method to other attributes, it would be necessary to have labelled data for each attribute of interest.

Supervised learning is a prevalent type of machine learning technique utilised for attribute extraction. Supervised learning techniques are typically used in scenarios in which there is a large amount of labelled data available to train a model to recognise patterns and make predictions regarding attributes. These techniques can achieve high accuracy in identifying attributes that may not be easily defined by rules. However, they typically require a significant amount of labelled data for training, which may not always be available, and can also be expensive. If we wish to have an approach that can be applied to any attribute, then the cost of data labelling would be rather high. In contrast, our proposed methodology offers an unsupervised approach for identifying demographic attributes of communities. This approach can be used to analyse any attributes and does not rely on labelled data.

More recently, researchers have ventured into the integration of neural network models for demographic inference. For instance, Vijayaraghavan et al. [91] developed a deep learning model by leveraging users' profile information, tweets, and images.

Wang et al. [95] explored the incorporation of profile-based features, such as usernames, screen names, biographies, and profile images, within deep learning models. Their study introduced a novel multimodal deep neural architecture for the simultaneous classification of age, gender, and organisation status (a binary organisation indicator "is-organisation," distinct from individual accounts) across social media users in 32 languages. Their method harnesses four sources of information: usernames, screen names, biographies, and profile images. Kim et al. [57] proposed graph-based recursive neural networks employing skip-gram embeddings. This model not only incorporates the user's text but also that of their network. It uses recursive neural networks to infer three demographic attributes of Twitter users: age, gender, and user type (individual, organisation or other). Liu et al. [52] delved into demographic inference on Twitter using text features in conjunction with various classic and deep learning models to infer gender and age. Classic models prove adequate for age inference but are overshadowed by deep learning models in gender inference. This research encompasses a broad spectrum of learning approaches, ranging from classic machine learning models to deep learning models, to elucidate the role of different language representations in demographic inference. Hiba et al. [30] proposed a deep learning approach for age estimation based on facial images. In parallel, Liu et al. [50] introduced a method that employs convolutional neural networks (CNNs) to capture word and sentence features related to age and gender. In addition, Liu et al. [51] presented a deep learning hierarchical network for inferring age exclusively from Twitter posts, specifically tweet text and emojis. This approach integrates independent linguistic knowledge obtained from text and emojis to make predictions. It is imperative to note that labelled data are required for training in the proposed approach.

Previous studies have primarily focused on identifying the attributes of individual users. In contrast, our study identifies a specific attribute among individual users and subsequently calculates the proportion of users who possess this attribute. Georgiou et al. [22] presented a method that can be used to determine if a hashtag group can actually be considered a community. First, they extract attributes of the participants such as location, age, gender or political affiliation. Then, they identify a certain community whose participants possess certain attributes, while those outside the community lack these attributes. These attributes are predominantly or exclusively present in the majority of participants. However, their method was not able to consider other attributes of interest. Moreover, Georgiou et al. used specific techniques to infer specific attributes. For example, they used a pattern-matching algorithm or Twitter's geotagging mechanism to extract locations [1] [90]; a language model [83] to extract gender and age; and users' communication with certain known accounts to determine political affiliation. The main difference between our work and that of Georgiou et al. is that their method is limited to extracting a set of specific attributes (location, age, gender or political affiliation), while ours is more general. They also categorise the community with specific attribute values. For example, to determine the gender of a community, they categorised it as either male or female as long as one is more dominant than the other. In contrast, we analysed the support for both males and females within a community. For example, our study reports that a community has 27% of male participants, 2% female, and the remaining participants were undetermined due to insufficient information present in the data. Another difference is that they first extracted attributes of participants and then used the attributes to ascertain if these participants form a community with certain characteristics. In contrast, we assume that we already have a community and need to determine the level of support for an attributes of interest within this community.

Table 2.1 provides a summary and classification of the reviewed papers on attribute extraction.

Table 2.1: The classification of attribute extraction papers

Ref No	**Data use**	**Technique use**	**Attributes**
[64]	Text-based	Rule-based	Gender

[76]	Text-based	Machine learning-based (supervised learning)	political affiliation and ethnicity
[2]	Text-based and network-based	Rule-based and machine learning-based (supervised learning)	Age, gender and political affiliation.
[83]	Text-based	Machine learning-based (supervised learning)	Age and gender
[102]	Image-based	Machine learning-based (supervised learning)	Age
[54]	Network-based	Machine learning-based (supervised learning)	Gender
[15]	Network-based	Machine learning-based (supervised learning)	Age, gender, ethnicity, education, income and child status
[85]	Text-based	Rule-based	Age, occupation and social class
[92]	Text-based and network-based	Machine learning-based (supervised learning)	Age and gender
[34]	Text-based and network-based	Machine learning-based (supervised learning)	Occupation
[67]	Text-based	Rule-based	Age
[91]	Text-based, network-based and image-base	Machine learning-based (deep learning)	Age, gender, political orientation and Location
[58]	Image-based	Machine learning-based (supervised learning)	Gender and race

[101]	Text-based	Machine learning-based (supervised learning and deep learning)	Age, gender and occupation
[57]	Text-based and network-based	Machine learning-based (deep learning)	Age, gender and user type
[98]	Text-based	Machine learning-based (supervised learning)	Age, gender and ethnicity
[89]	Text-based and image-base	Machine learning-based (supervised learning)	Gender
[95]	Text-based and image-base	Machine learning-based (deep learning)	Age, gender and organisation status
[35]	Image-based	Machine learning-based (supervised learning)	Age, gender and ethnicity
[99]	Text-based	Rule-based	Ethnicity
[52]	Text-based	Machine learning-based (supervised learning and deep learning)	Gender and age
[30]	Image-base	Machine learning-based (deep learning)	Age
[51]	Text-based	Machine learning-based (deep learning)	Age
[50]	Text-based	Machine learning-based (deep learning)	Gender and age
The proposed approach	Text-based	Unsupervised learning	Any attribute

2.3 Semantic Analysis

In this section, we discusses semantic analysis for text similarity. We review typical methods for deriving semantic similarity, knowledge-based similarity and distribution-based similarity.

2.3.1 Overview of Semantic Similarity

Measuring text similarity is a fundamental topic in natural language processing research and applications. Text similarity measurements are frequently used to assess the similarity of words, phrases or documents, and they can be broadly divided into three categories: syntactic [9, 93], semantic [36] and hybrid methods. The syntactic similarity is calculated by counting string matching, word order or words co-occurrence. For example, if two texts have the same set of words, then they would be considered similar. Semantic similarity refers to closeness in meaning [48]. For example, consider the following two sentences: 'Sara invited David to dinner', and 'David invited Sara to dinner'. Because they have the same word set, these two phrases are considered to be similar (in fact identical) in terms of syntactic similarity. However, they are fundamentally different in terms of semantic similarity. Despite the resemblance of the word set, they have different meanings. Both semantic and syntactic are used in hybrid methods. They integrate syntactic analysis, which examines word structure and arrangement, with semantic analysis, which considers word meaning and context. This combination results in improved text similarity assessments.

In this thesis, we utilise semantic similarity to measure the similarity between user profiles and attributes based on their underlying meaning and concepts rather than their syntactic likeness. By comparing the semantic content of a profile with an attribute of interest, we can determine whether a profile possesses a particular attribute. Semantic similarity enables us to better match a profile against an attribute, thereby enabling more accurate assessments of attribute presence or absence. Semantic similarity among words, texts and documents is actively researched in a variety of fields, including artificial intelligence, natural language processing, the semantic web and semantic search engines. It is used in applications such as answering questions [72, 27, 21], query expansion [24], plagiarism detection [40], automatic text summarisation [96, 78], semantic search [70, 81], document classification [14, 29] and many more.

There are various techniques for semantically determining word similarity [100, 42, 44, 31] and sentence similarity [59, 45, 36, 79, 20]. These techniques include cosine similarity, Jaccard similarity, and edit distance. Cosine similarity evaluates the angle between two words frequency vectors, whereas Jaccard similarity quantifies the overlap between two sets of words. On the other hand, edit distance computes the minimum number of operations, such as insertions, deletions and substitutions, necessary to transform one text string into another.

The techniques employed to semantically measure word or sentence similarity can be broadly categorized into two approaches: knowledge-based approaches and distributional approaches. Knowledge-based approaches represent words and concepts by utilising structured knowledge resources like ontologies or semantic networks. These resources organise information based on semantic relationships, such as hierarchical connections like hypernymy (is-a relationship) or meronymy (part-whole relationship). For example, in an animal ontology, “cat” can be a subclass of “mammal”, and “mammal” can be a subclass of “animal”, thereby reflecting their hierarchical relationships. Distributional-based approaches assume that words with similar meanings are likely to appear in similar contexts. This assumption is used to create word embeddings or vector representations that capture the semantic relationships between words based on their co-occurrence patterns in a large text corpus. For example, by training a word embedding model like Word2Vec on a large collection of news articles, the model can learn to represent words as dense vectors in a high-dimensional space. Words that frequently occur in similar contexts, such as “car” and “vehicle”, would have similar vector representations, thereby reflecting their semantic similarity.

2.3.2 Knowledge-based Approaches

Knowledge-based approaches utilise a semantic network of words that encompasses the meanings of words and the relationships between them. These relationships are typically coded in a knowledge source, such as WordNet, which provides a hierarchical structure of concepts and the relationships between them. The similarity between two words in this approach is based on their connections in the knowledge source. For example, the semantic similarity between `girl` and `female` can be determined by finding the shortest path between them in WordNet. The phrase “shortest path” describes a path with the fewest edges or semantic connections between two terms.

The knowledge-based approach is divided into two categories: gloss-based, and feature-based models. Gloss-based models, such as the one proposed by Lesk [43], leverage the definitions (glosses) of words in a dictionary to measure their similarity. The Lesk algorithm assesses the degree of overlap between the glosses of two words and generates a similarity score based on the count of shared words. Feature-based models, such as WordNet [62], represent words as sets of features, such as hypernyms (superordinate concepts), hyponyms (subordinate concepts), or synonyms, and then compute the similarity between words based on the overlapping of their feature sets. In feature-based models, according to Lyons [56] and other structural linguists, words cannot be defined independently of other words. Thus, the meaning of a word is influenced by its relationships with other words.

Several researchers have examined the use of semantic relationships to improve information retrieval. Information retrieval is the process of identifying pertinent information from a vast collection of unstructured or semi-structured data such as text, images, audio, video, and web pages. For example, Hassanein et al. [29] utilised semantics with social network platform data, particularly Facebook status updates, to predict Big Five personality traits: extraversion, agreeableness, conscientiousness, neuroticism, and openness. The WordNet dataset is used to determine the semantic similarity between user-posted text and the terms that describe the personality trait. ComQA is a framework developed by Jin et al. [37] that allows end-users to ask complex questions and receive answers. Answering questions is a form of information retrieval that answers a user's question with facts or text excerpts retrieved from documents. This necessitates determining explicit semantic relationships between document ideas and concepts in the user's query. The ComQA framework includes a three-phase knowledge-based question-answer process. A complex question is broken down into numerous triple patterns in ComQA. Then, ComQA searches the knowledge base for possible subgraphs that fit the triple patterns. Thereafter, to discover the answer, it evaluates the semantic similarity between the subgraphs and the triple patterns. Mohamed and Oussalah [66] suggested an approach to identifying paraphrasing. When sentences contain a collection of named-entities, the approach solves the problem of examining sentence-to-sentence semantic similarity. The similarity is calculated by combining word semantic similarity derived from WordNet taxonomic relationships with named-entity semantic relatedness acquired from Wikipedia.

These are some examples of the approaches that have been proposed for identifying semantic relationships using knowledge-based methods, among many others that exist in the literature. In our work, we utilize semantic relationships to expand the attribute word to identify words that are related to it in meaning. For example, hyponyms is a type of semantic relationship that represents specific examples or subcategories of the given word and can be used to provide a more specific context. For example, if the given attribute is `religion`, hyponyms such as `Christianity`, `Islam`, and `Hinduism` can be used as expanded words for religion.

2.3.3 Distributional Approaches

Distributional similarity involves representing words as vectors based on their distributional properties in a corpus of text. This approach is based on the distributional hypothesis, which suggests that words that occur in similar contexts tend to have similar meanings. A co-occurrence matrix is often used to represent the distributional properties of words. This matrix records the frequency of co-occurrences between words. Then, a vector space model is constructed, where each word is represented as a vector in a high-dimensional space. The similarity between words is measured based on the distance between their vectors, using cosine similarity. The closer the cosine value is to 1, the more similar the words are considered to be.

Distributional models can be categorized into two main types: count-based models and predictive models. Count-based models, such as Latent Semantic Analysis (LSA) [17], utilize a co-occurrence matrix to represent the distributional characteristics of words in a corpus. The matrix is constructed by recording the frequency of word co-occurrences and then decomposing it using Singular Value Decomposition (SVD) to obtain low-dimensional representations of words. Predictive modeling is used to learn word embedding, a sort of dense vector representation for words. It involves representing words as high-dimensional vectors based on their distributional properties in a corpus of text, with the goal of capturing both syntactic and semantic relationships between words. Syntactic relationships involve the arrangement and roles of words within a sentence, while semantic relationships focus on the meaning and interpretation of those words within the context of the sentence. Word2vec [60] and GloVe [77] are two examples of successful word embedding implementations that employ neural networks and matrix factorisation, respectively, to learn embedding vectors.

Word2Vec is a popular technique for generating word embeddings by training on word sequences. Word2Vec models are trained based on the distributional hypothesis, which posits that words appearing in similar contexts are likely to have similar meanings. However, this approach has limitations in accurately capturing the nuances of word relations, such as synonymous terms and hierarchical links [26]. Word2Vec treats words as isolated entities, thereby disregarding the specific contexts in which they occur and the potential variations in meaning across different contexts.

A number of researchers have proposed methods for calculating semantic similarity that relies solely on distributional approaches. Tom [39] calculated the similarity between short texts using a pre-trained word vector, with text represented by the average vector of words' vectors. A semantic similarity is derived using the averaged vectors. TF-IDF is also utilised in the similarity equation to weight the words in the sentence according to their importance. This method does not utilise other resources like WordNet because external resources are not available across all areas and natural languages. Pawar and Mago [74] proposed an unsupervised approach to learn the semantic similarity between words and phrases. This approach merges distributional semantics with graph-based approach. In the graph-based approach, words are represented as nodes and their relationships as edges in a graph; the algorithm incorporates contextual and structural information from word embeddings, thereby enabling a more comprehensive representation of word relationships. Consequently, the proposed graph-based method enhances the accuracy of measuring semantic similarity. Shao et al. [84] proposed a transformer-based neural network for answer selection. Their approach attempts to extract both global and sequential information from question and answer sentences. They begin by using a serial structure to implement a multi-head self-attention mechanism and a BiLSTM as a feature extractor. They also use three aggregated techniques in the relevance matching layer to pool the sentence representation matrix into a sentence embedding. Jin et al. [38] proposed a method for calculating the similarity in meaning between words, which relies on the Word2Vec technique. To calculate the semantic similarity of words, this method combines a semantic dictionary and large-scale corpus statistics, as well as a weighting strategy.

Our work utilizes distributional methods to measure semantic similarity between profiles and expanded words associated with attributes obtained through knowledge-based methods. The calculated similarity scores indicate the extent to which an attribute is present in a profile. The most closely related work to our research in the field of semantic similarity is the one proposed by Patil and Ravindran [73]. They proposed an approach for classifying software defect reports that reduces the need for labelled training data. Their method relies on measuring the semantic similarity between a given software defect report and the textual descriptions of known defect types. They first created representations of a software defect report and software defect types by projecting their textual descriptions into a concept-space that was spanned by Wikipedia articles, using the Explicit Semantic Analysis (ESA) framework developed by Gabrilovich and Markovitch in 2007 [19]. Then, they calculated the semantic similarity between these representations and assigned the software defect type that had the highest similarity to the defect report. This approach enabled them to effectively classify software defects with limited labelled data.

2.4 Summary

In this chapter, we discussed some background information on communities. Then, we described the methods of attribute extraction. Finally, knowledge-based and distributional-based approaches to establishing semantic similarity of terms were reviewed.

The aim of this thesis is to determine the characteristics of people. A number of approaches have been proposed to extract attributes. However, these approaches are largely based on supervised learning and, as such, they require a large amount of labelled data. In the next chapter, we propose an unsupervised approach to extract any attribute for a community.

Chapter 3

Lexicon-Based Attribute Extraction

Attributes extraction is a often required to characterise community members. Existing approaches to attributes extraction rely primarily on supervised learning, which requires a large amount of data to be labelled manually. Furthermore, while these methods are able to extract certain attributes rather well, their effectiveness is limited to those specific attributes only and they are not useful for extracting other attributes.

In this chapter, we first present our two-stage methodology for extracting attributes from hashtag communities—lexicon-based attribute extraction (LBAE) stage and classification enhanced attribute extraction (CEAE) stage. We then explain the LBAE stage in detail, where lexical sources and semantic analysis are used to determine whether members of a #hashtag community have a particular attribute of interest. Our approach differs from previous methods in that it is unsupervised; therefore, is more efficient and applicable to any attribute of interest. Part of the work presented in this chapter has been previously published in [3]

This chapter is structured in the following manner: In Section 3.1, we define the problem of attributes extraction, and discuss the two general approaches to solving this problem (bottom-up derivation and top-down derivation). Then, Section 3.2 provides an overview of our two-stage methodology. Section 3.3 describes the LBAE stage in detail, and Section 3.4 concludes the chapter.

3.1 Problem Definition

Before explaining the proposed method, it is useful to formally define the problem of extracting attributes from user profiles for a community first.

Definition 1 (Community) *: A community $\mathcal{C}$ is represented by $\mathcal{C} = (U, P)$, where $U = \{u_1, u_2, \ldots, u_n\}$ is a set of users (people) who participate in the community and $P = \{p_1, p_2, \ldots, p_n\}$ is a set of profiles, each associated with one user.*

Profiles associated with users are typically written as free text. For our work, we assume that the textual profiles have already been converted into term vectors. In other words, a profile p is represented as $\langle t_1, t_2, \ldots, t_k \rangle$, where each t_i is a term (literal) extracted from p.

Definition 2 (Person Characteristic) *: A person characteristic is an attribute-value pair (A, v), where A is a literal representing a characteristic type that describes a user and v is a literal representing an instance of A.*

For example, `Hobby` is a characteristic type and `swimming` is a value of this type. Thus, (`Hobby, swimming`) represents a person characteristic. Note that a person may have multiple values for the same characteristic type. For example, a person may have (`Hobby, swimming`) and (`Hobby, reading`). As a shorthand, we allow these to be written as a set in a person characteristic: (`Hobby, {swimming, reading}`). In certain instances, when providing context, we may refer to the person characteristic simply as an attribute.

To search for a person characteristic, we search for an attribute with specific value (A, v) or an attribute with any value $(A, *)$. For example, we could search for `(Religion,*)`, which would result in users who have any `Religion` values such as `Islam`, `Christianity` or `Buddhism`. On the other hand, if we want to search for users with `Christianity` as `Religion`-that is `(Religion,Christianity)`-then only the users who have the `Christianity` value will be returned.

Definition 3 (Attribute Extraction) *: Attribute extraction is the process of identifying whether a user has a certain person characteristic* (A, v)*. The input is a community* $\mathcal{C} = (U, P)$ *and a desired person characteristic* (A, v)*, and the output is the proportion of users possessing the desired person characteristic* (A, v)*.*

For example, given a community $\mathcal{C} = (U, P)$ *and a desired person characteristic* $(Gender, Female)$*, using profiles texts of users in this community to identify if these users can be considered as female, and produces an output, which is the proportion of users possessing this attribute.*

3.1.1 #Hashtag Communities

The definitions given in the previous sections are sufficiently generic for any type of online community, as long as each community is characterised by a set of users and a set of profiles describing them. In this section, we specifically consider the communities formed around #hashtags in Twitter. When a group of Twitter users tweet on the same #hashtag, we say that this group of users forms a community and we call it a *#hashtag community*, denoted by $\mathcal{C}_{\#hashtag}$, and its size is the set of distinct users tweeting on the #hashtag, denoted by $|\mathcal{C}_{\#hashtag}|$.

From the community given in Table 1.1, it is evident that words such as `Jesus`, `catholic` and `christian` appearing in the profiles of u_1, u_2, u_3 and u_4 should suggest that they are religious and, more specifically, have the faith of Christianity. Equally, words such as `mom` and `mother of 6` appearing in the profiles of u_1 and u_7 should enable us to infer another person characteristic `(Gender, female)` for these users.

3.1.2 Bottom-up Derivation

The bottom-up derivation is a *search* based approach to person characteristic derivation. The main steps of this approach are presented in Figure 3.1. We begin with user profiles and attempt to extract relevant values of person characteristics and then associate them with the types. For example, considering the profile in Table 3.1, we can extract words such as `husband`, `swimming`, and `Christian`. We can then associate these words with corresponding types, such as associating `husband` with `Married`, `swimming` with `Hobby`, and `Christian` with `Religion`.

Table 3.1: An example of a profile

User (U)	Profile (P)
u_1	Prostate cancer survivor, husband, swimming enthusiast, father, devout Christian, leadership survivor, churchgoer, grateful for Aba Father, love Jesus, The Holy Spirit & Christianity

While this approach is desirable in that it is rather general and is not limited by which person characteristics may be extracted, it is not a trivial task and involves

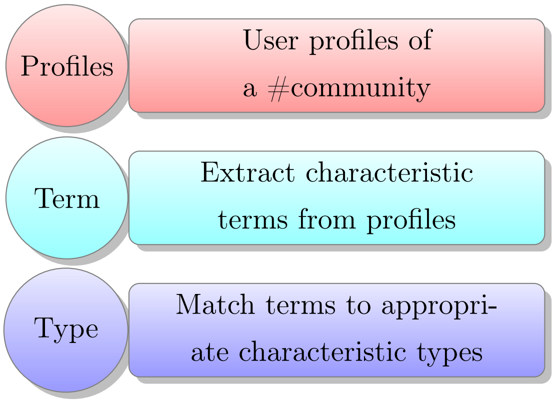

Profiles
User profiles of a #community
Term
Extract characteristic terms from profiles
Type
Match terms to appropriate characteristic types

Figure 3.1: Bottom-up derivation

two major challenges.

- First, it must be determined which term(s) or word(s) occurring in a user profile are meaningful characteristics values. For instance, in our example in Table 1.1, it is not straightforward to decide that terms in the profile of u_1 such as `mom`, `catholic` and `Jesus` are relevant person characteristics values, whereas `addicted` and `proud` are not useful. One possible solution is to employ machine learning techniques to identify such terms, but this would require a large amount of annotated training examples, which can be difficult to obtain.

- Second, assuming that we are able to obtain a list of meaningful characteristic values, determining their types is difficult. For instance, in our example, linking values such as `Catholic` and `Christian` to the possible `Religion` type is difficult. This would require a substantial knowledge base, either constructed as a dictionary or derived from machine learning, which could be difficult to obtain. Moreover, some of the terms may have multiple meanings; for example, `author` could indicate a type of `occupation` or a kind of `hobby`, thereby adding further complexity to this approach.

3.1.3 Top-down Derivation

The top-down derivation is a *detection-based* approach to the derivation of person characteristic. The main steps of this approach are illustrated in Figure 3.2. We begin with a characteristic type given by the user and attempt to detect values in user profiles that are relevant to the given type. For example, suppose that we would like to identify if a given profile contains (Religion, Christianity). We can attempt to detect values relevant to Christianity within the profile in Table 3.1. In analyzing this profile, we find that the terms Jesus, Christian, and Christianity are closely associated with Christianity; therefore, we can decide that this profile includes Christianity.

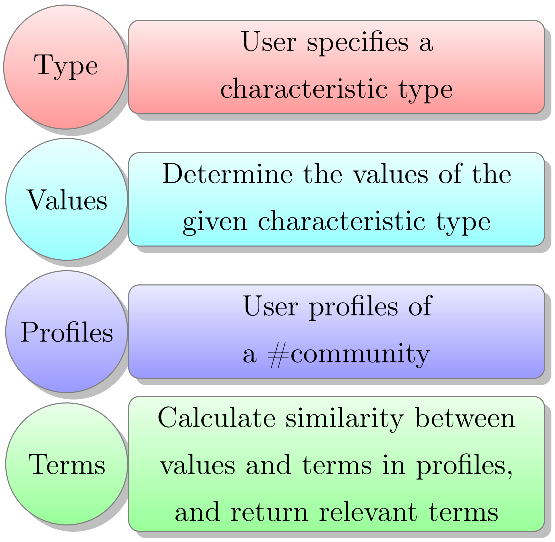

Type
User specifies a characteristic type
Values
Determine the values of the given characteristic type
Profiles
User profiles of a #community
Terms
Calculate similarity between values and terms in profiles, and return relevant terms

Figure 3.2: Top-down derivation

This approach is clearly less general than the bottom-up approach, as it can only detect whether a particular person characteristic exists among the profiles and requires all possible person characteristics that may be present in the community to be specified. However, the ability to detect any given person characteristic type is still useful, as it can help monitor characteristics of a community. For example, if we wish to determine whether a #hashtag community is associated with a specific type of `Religion`, then we can search through all profiles of the users in this community to see if they contain that value.

The top-down approach does not include the difficult tasks of determining relevant values from profiles and then mapping these values to the correct types like the bottom-up approach has to deal with. However, it still has some substantial challenges to address.

- First, we need to have a set of values associated with a given characteristic type, so that we can use them to search through user profiles to determine whether the given type is supported by the community. For example, when a given characteristic type is `Religion`, we need the values of `Muslim`, `Christian`, `Worship` and many more to search through the user profiles and understand whether the community can be characterised as religious. One possible solution is to use a dictionary containing possible values for each characteristic type, for example, using Wordnet or ConceptNet. Alternatively, more dynamic semantic tools such as word embeddings may be used to determine the associations of any value to any characteristic types.
- Second, we need to consider how to effectively and accurately count the support for a given characteristic type. For instance, in our example, `mother` and `mom` are two different but semantically equivalent values of a characteristic type such as `parent`. It is necessary to combine them when counting the support for this characteristic type. The solution to the first task above partially addresses this issue, that is `mother` and `mom` can both be values of `parent`; hence, they will automatically be included in the counting for `parent`. However, Twitter profiles can include abbreviations, spelling errors, slang, etc., which pose addi-

tional challenges for determining semantic equivalence between values. Again, techniques such as word embeddings or machine learning could be employed to address this issue.

In this thesis, our objective is to extract attributes from community members that users are interested in. To achieve this goal, we adopted the detection-based approach. Specifically, our approach involves beginning with a desired person characteristic provided by the user and identifying a set of values associated with it. We then proceed to calculate the extent of semantic equivalence between the identified values and the profiles of community members to determine the proportion of profiles that possess the desired attributes.

3.2 Proposed Approach

In this section, we present an overview of our approach to extracting attributes from hashtag communities. Our approach intends to eliminate the need for manual labelling. We leverage semantic relationships and similarity to automatically extract attributes from user profiles. As such, we eliminate the requirement for manual labelling, thereby lowering the possibility for bias and human error and making the process scalable to handle any attributes. Prior research has employed similarity-based methods to extract attributes from textual data. For example, Patil and Ravindran [73] employed semantic similarity to classify software defect reports by comparing the textual descriptions of known defect types with a given software defect report. Similarly, Hassanein et al. [29] used semantic similarity between user-posted text and personality trait terms to predict personality traits.

Our methodology consists of two stages: the LBAE and CEAE stages. Figure 3.3 presents an overview of our methodology. Our methodology yields two status of profiles: classified profiles and unknown profiles. Classified profiles are profiles whose attributes have been identified or assigned. These attributes can range from a single attribute to multiple attributes. On the other hand, unknown profiles are those for which the presence or absence of attributes remains unknown.

The LBAE stage. In this stage, we use an unsupervised technique to extract attributes. This is because it is difficult to label a large amount of data for supervised learning, particularly if we want our method to work with any attribute. This stage depends on a semantic analysis to determine whether the profile has the attribute of interest, and it consists of two steps, as depicted in Figure 3.4.

In the first step, we use a lexical source to expand a person characteristic given by the user. Generically, given a single word, we find its semantically equivalent words, which can then be utilised to search for the person characteristic. For example, suppose that an attribute that the user wants to determine in a community is `(Religion, Christianity)`. We expand `Christianity` to include terms such as `Christianity`, `Adventism`, `Albigensianism`, `Catholicism`, `Donatism`, `Protestantism`, `Tractarianism`, and `Puseyism`, which are obtained by utilising a lexicon source such as WordNet. People often express a single concept or idea in various ways. When we search for an attribute in a profile, we provide a single word representing that attribute. This word has synonyms or related terms with similar meanings. By utilizing these terms, we increase our likelihood of effectively matching the attribute.

In the second step, we compute the similarity between the expanded words and terms that occurred in a user profile to determine whether or not the user has the attribute.

Figure 3.5 illustrates an example of determining if a user's profile can be identified as including `(Religion,Christian)`. Assuming that the profile has been converted into a term vector already, each word in the set of expanded words is compared to

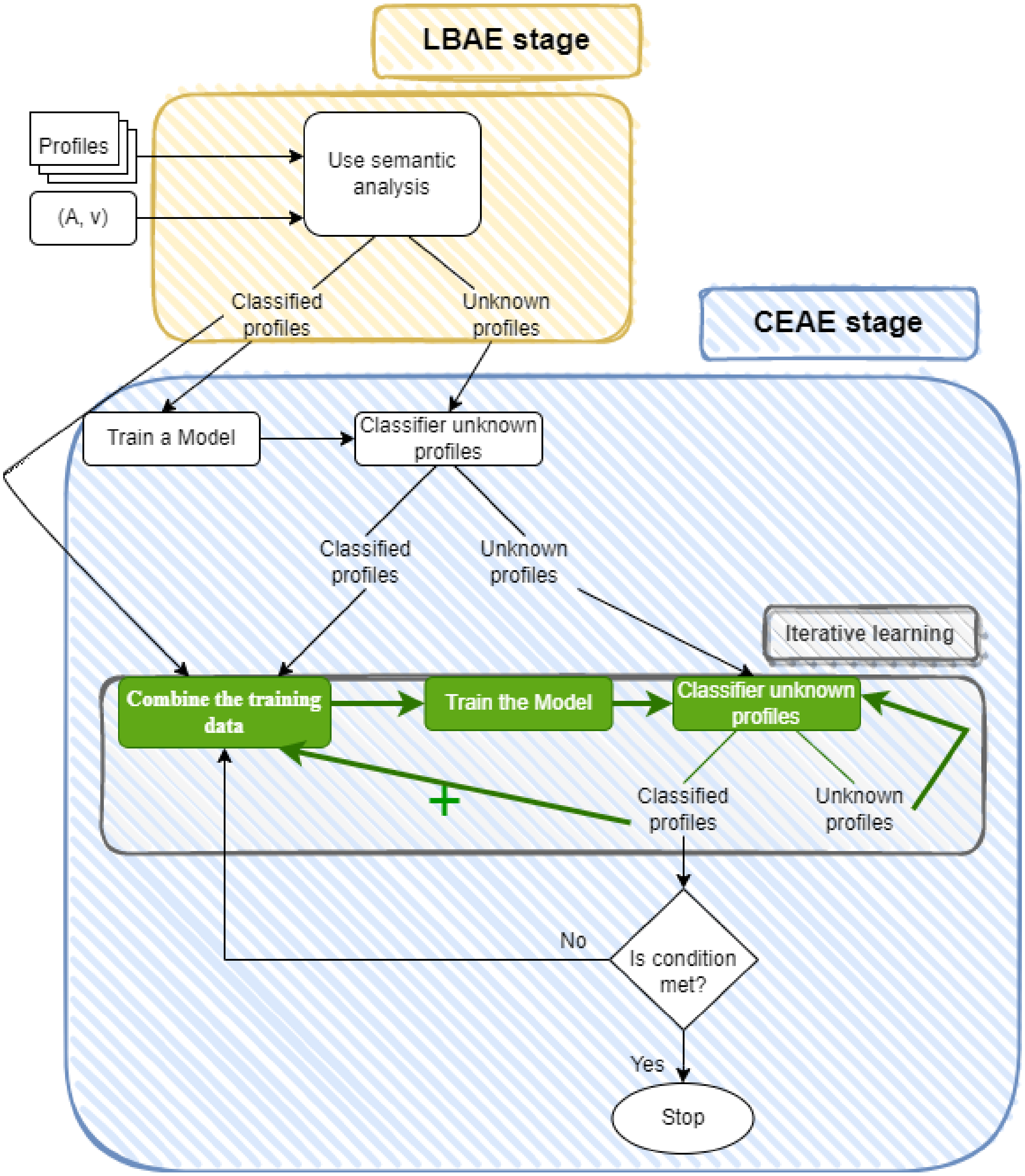


Figure 3.3: An overview of the proposed methodology

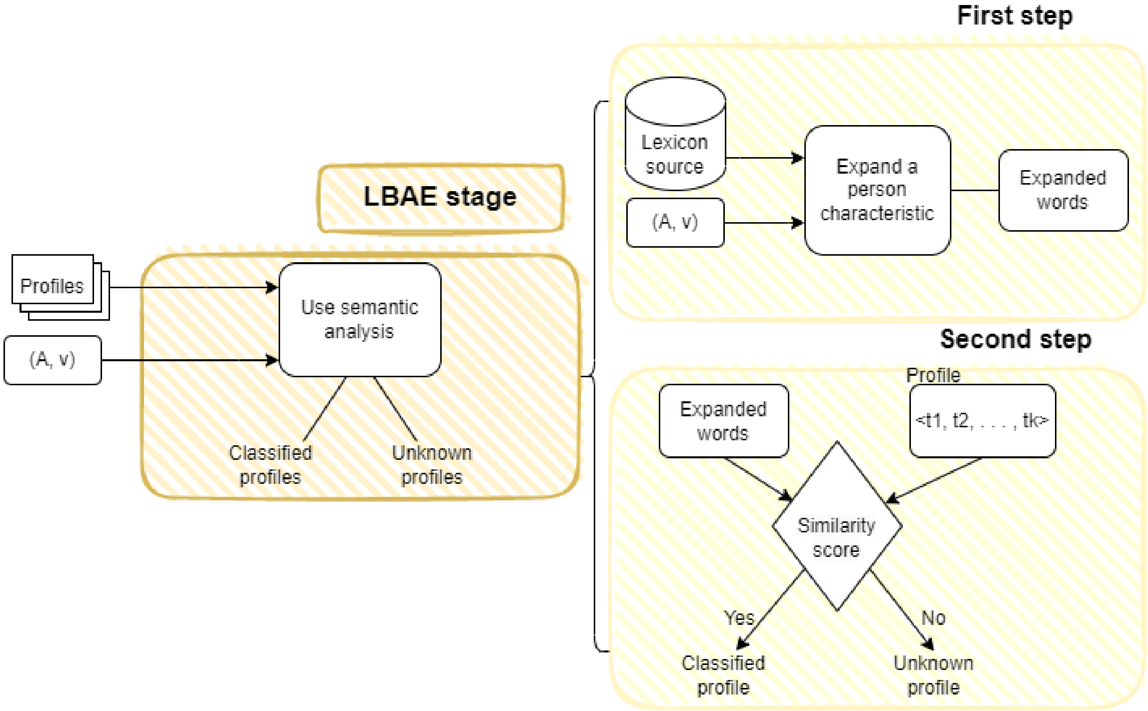


Figure 3.4: An overview of the LBAE stage

each term in the profile vector. In Table 3.2, suppose we have these words and use Word2Vec to calculate the similarity score between extended words and words in the profile of user U_1. Suppose that the two terms are deemed to be similar if their similarity score is 0.50 or higher, then `catholic` and `Jesus` in the profile are similar to the expanded words `Catholicism` and `Christianity`, respectively; hence, U_1 is considered to have `(Religion,Christian)`. In contrast, the attribute of the profiles of users U_2 and U_3 are considered to be unknown, as the similarity scores are too low. Note that although the profile of U_2 may be related to `Christianity`, as the user profile includes terms such as `believe` and `church`, the LBAE stage cannot determine the attribute for this profile since the similarity scores between its terms and expanded words are insufficient.

While the LBAE stage can classify profiles without labelled data, there can be pro-

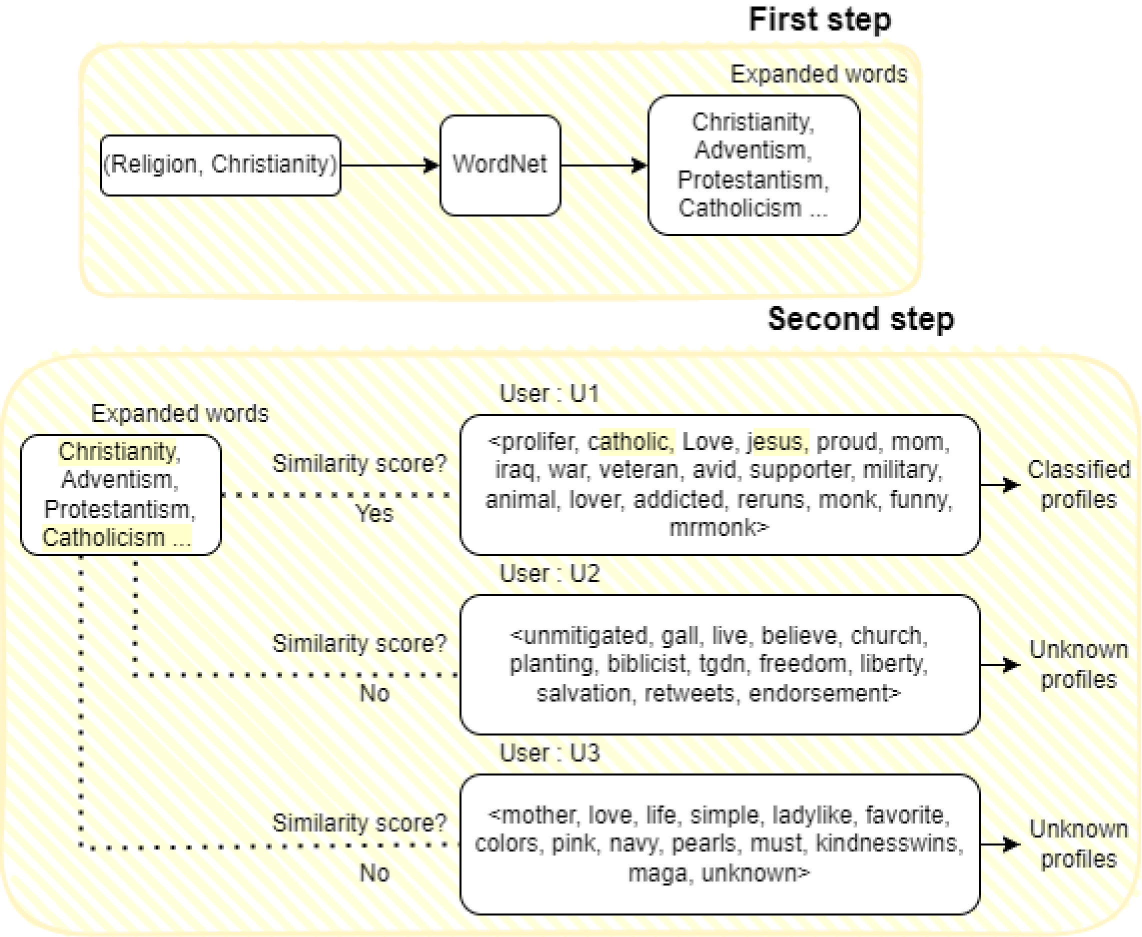

First step
Expanded words
(Religion, Christianity)
WordNet
Christianity, Adventism, Protestantism, Catholicism ...
Second step
Expanded words
Christianity, Adventism, Protestantism, Catholicism ...
Similarity score?
Yes
User : U1
<prolifer, catholic, Love, jesus, proud, mom, iraq, war, veteran, avid, supporter, military, animal, lover, addicted, reruns, monk, funny, mrmonk>
Classified profiles
Similarity score?
No
User : U2
<unmitigated, gall, live, believe, church, planting, biblicist, tgdn, freedom, liberty, salvation, retweets, endorsement>
Unknown profiles
Similarity score?
No
User : U3
<mother, love, life, simple, ladylike, favorite, colors, pink, navy, pearls, must, kindnesswins, maga, unknown>
Unknown profiles

Figure 3.5: An example of the LBAE stage

files whose attribute cannot be determined during this stage due to the manner in which profiles are written. For example, some profiles have a given attribute but their writing style ultimately resulted in being different. Twitter profiles contain abbreviations, spelling errors, or slang, which can make it challenging to determine semantic equivalence. For example, `BRO` as an abbreviation of `brother` is commonly used to refer to a `male friend`, `BF` for `boyfriend`, `GRL` for `girl`, and `FEM` for `female`. Another problem that can arise is when profiles are too short. If the profile is not sufficiently long, it becomes challenging to extract attributes from it. To address

Table 3.2: An example of semantic similarities between the profile of U_1 and the expanded words for Christianity

		Expanded words			
		Christianity	Adventism	Protestantism	Catholicism
Profile terms	catholic	0.49	0.35	0.44	0.61
	love	0.22	0.10	0.14	0.21
	jesus	0.51	0.18	0.18	0.22
	proud	0.02	0.01	-0.01	0.02
	mom	0.10	0.06	0.03	0.13

these issues, we propose the CEAE stage, which involves training a classifier using the results from the LBAE stage as labelled data, and using the trained classifier to determine the profiles that we are unable to decide during the LBAE stage.

The CEAE stage. In this stage, we build a classifier using the data from the LBAE stage, and then the model is used to classify profiles with undecided attributes from the LBAE stage. To maximise the classification of the profiles, we build the classifier incrementally or iteratively, as explained below.

Figure 3.6 illustrates an example of the CEAE stage. Suppose the user wants to extract (Religion, Christianity) from profiles. The profiles of U_1 and U_2 have been classified as (Religion, Christianity) in the LBAE stage, whereas the profiles of U_3 and U_4 are left unknown in the LBAE stage. The classifier is trained using the profiles of U_1 and U_2, thereby assuming that the classifier learns based on frequency, where the frequent appearance of the word church leads to it being associated with Christianity. This new discovery, learnt from the data generated from the LBAE stage, will then allow U_3 to be classified as including (Religion, Christianity).

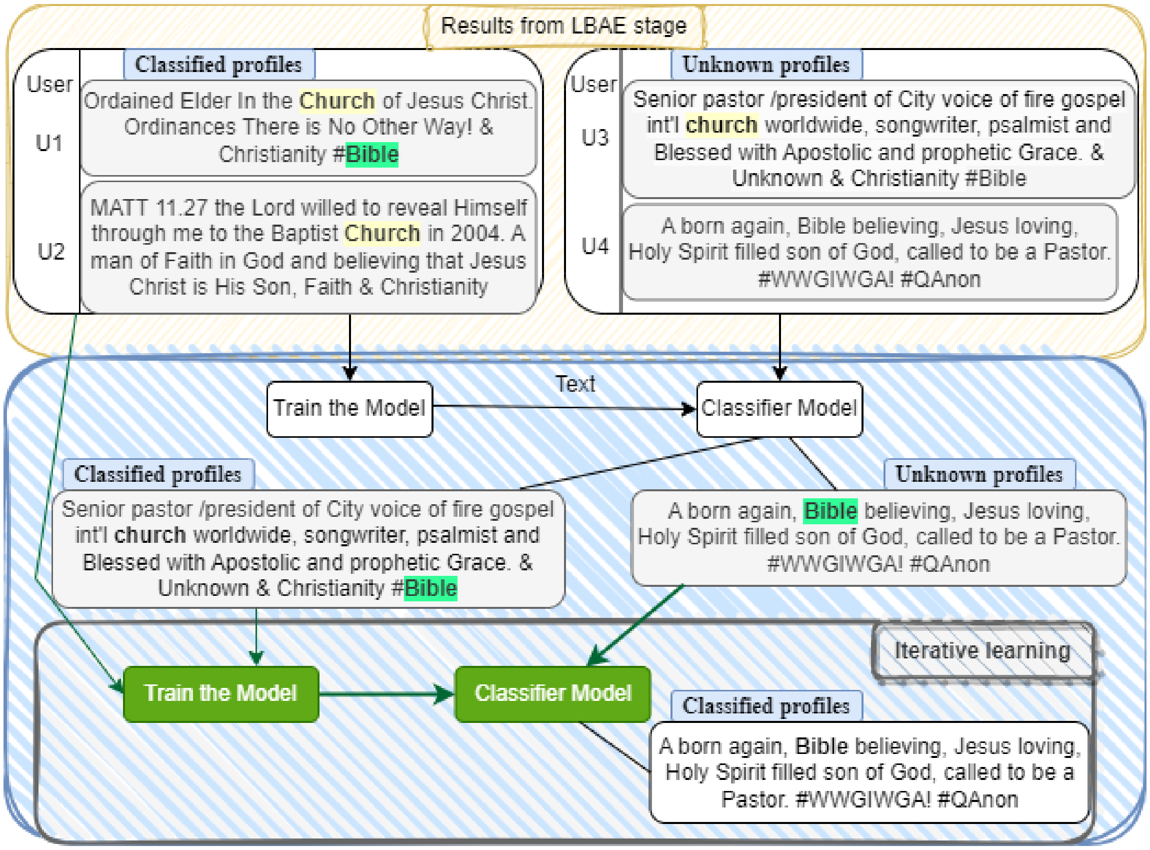

Results from LBAE stage
User
U1
U2
Classified profiles
Ordained Elder In the Church of Jesus Christ. Ordinances There is No Other Way! & Christianity #Bible
MATT 11.27 the Lord willed to reveal Himself through me to the Baptist Church in 2004. A man of Faith in God and believing that Jesus Christ is His Son, Faith & Christianity
User
U3
U4
Unknown profiles
Senior pastor /president of City voice of fire gospel int'l church worldwide, songwriter, psalmist and Blessed with Apostolic and prophetic Grace. & Unknown & Christianity #Bible
A born again, Bible believing, Jesus loving, Holy Spirit filled son of God, called to be a Pastor. #WWGIWGA! #QAnon
Train the Model
Text
Classifier Model
Classified profiles
Senior pastor /president of City voice of fire gospel int'l church worldwide, songwriter, psalmist and Blessed with Apostolic and prophetic Grace. & Unknown & Christianity #Bible
Unknown profiles
A born again, Bible believing, Jesus loving, Holy Spirit filled son of God, called to be a Pastor. #WWGIWGA! #QAnon
Iterative learning
Train the Model
Classifier Model
Classified profiles
A born again, Bible believing, Jesus loving, Holy Spirit filled son of God, called to be a Pastor. #WWGIWGA! #QAnon

Figure 3.6: An example of the CEAE stage

However, the profile of U_4 is still not classified at this time. However, since the profile of U_3 is now classified, we combine it with the profiles of U_1 and U_2 in the next round of training, thereby building a new classifier to classify U_4. The presence of the word `bible` in the profiles of U_1 and U_3 now suggests that `Christianity` and `bible` are associated. As a result, the profile of U_4 is classified as including `(Religion, Christianity)`.

One issue of the proposed iterative classification is to determine when the iteration must end. The termination of classification can be based on a user-specified condition. One option is to end the process after a specified number of iterations, which provides simplicity. An alternative strategy is to terminate the iterative classification when a specific convergence criterion is satisfied. For example, termination can occur when a certain proportion of profiles are classified. Granting users the flexibility to determine the stopping criteria for iterative classification is beneficial because different applications have different requirements and priorities. Some may prioritize accurate attribute extraction, while others prioritize the number of profiles being classified. Therefore, allowing users to specify the stopping criteria can allow them to cater to their specific needs.

It is worth noting that while our approach utilises a supervised classifier building process, there is no need to label the data. The labelled data are obtained from the LBAE stage. Thus, our approach remains an unsupervised method.

3.3 Lexicon-Based Attributes Extraction

We now describe the LBAE stage in detail and discuss the CEAE stage in Chapter 4. In the LBAE stage, we extract attributes by using a lexical source and semantic analysis. The LBAE stage comprises two primary steps: candidate value generation and similarity score calculation. In the candidate value generation step, candidate values of a person characteristic are obtained using a lexicon knowledge source. In the similarity score calculation step, the similarity between candidate values and words in profiles is calculated. In Sections 3.3.1 and 3.3.2, we present these two steps in detail.

3.3.1 Candidate Values Generation

The reason for expanding an attribute is to expand the search space and potentially discover additional relevant profiles. These expanded words are referred to as candidate values. For example, when we extract `Christianity` from profiles, we can include relevant terms such as `Jesus` and `Catholicism` to encompass different words used to describe the same attribute. This approach enables us to extract the attribute from potentially more relevant profiles. Considering candidate values for the attribute enhances the effectiveness of our attribute extraction.

In this section, we utilise a knowledge source such as WordNet or ConceptNet to find a set of candidate values for a given person characteristic. Knowledge-based approaches organise terms in a hierarchical manner, and the relationships between the words are represented by a number of relational descriptors, such as synonyms and hyponyms. Knowledge-based methods are beneficial when applications need to encode hierarchical interactions among words [6].

Suppose that we are given a person characteristic (A, v). We expand this person characteristic to $\langle cv_1, cv_2, \ldots, cv_f \rangle$, where each cv_i is a value that is semantically related (e.g. synonyms or hyponyms) to v in (A, v) or A in $(A, *)$. For example, if the given person characteristic is `(Religion,Christianity)`, then `Christianity` would be expanded. On the other hand, if it is `(Religion,*)`, then `religion` will be expanded. In (A, v), the focus is on extracting A specifically related to a particular value represented by v. Here, v acts as a qualifier that narrows down the attribute extraction to specific value. On the other hand, in $(A, *)$, the $*$ is a wildcard symbol that represents a broad extraction. It implies that no specific value is being specified. The two different ways, (A, v) and $(A, *)$, enable flexibility in attribute extraction. (A, v) is useful when we have a specific value in mind related to A that we want to focus on. $(A, *)$ is beneficial when the aim is to conduct extraction of various values of A without a specific value in mind.

It is crucial to maintain a balance in the extent of expansion. When the extension covers too many specific words, while there is potential for increased classification, accuracy could suffer. This is because of the possibility that more specific words could occasionally introduce noise and they might not be relevant to the context. Conversely, limiting the expansion to the most related words can result in good accuracy but lower classification. The goal is to achieve both accuracy and a high number of classified profiles, but this may not always be possible. Therefore, it is important to strike a balance between the number of classified profiles and the accuracy of the classification.

There are no universally candidate values; the candidate values will depend on the lexicon source used. We employ WordNet (WN) as a knowledges source to obtain the candidate values. WordNet was selected because it is a popular lexical source and not because it is necessarily the best. Our purpose is to demonstrate the worth of our two-stage approach, not to demonstrate that WordNet is the best. Using a better source than WordNet will surely enhance our results. In addition, we use ConceptNet to find the candidate values. ConceptNet is an open-source semantic network that was developed to help computers understand word meanings. ConceptNet is used to demonstrate how different sources, such as WordNet and ConceptNet, have an impact on the results.

WordNet

WordNet [62] is an English lexical database that is commonly used to link words into semantic relations, including synonyms and hyponyms. Like a traditional dictionary, WordNet provides definitions of terms as well as their relationships. However, WordNet differs from a normal dictionary in that it is organised conceptually rather than alphabetically. Moreover, WordNet has many features, including synset, hypernym and hyponym. Synset instances are a collection of synonyms that communicate the same idea. Words in WordNet are organised in a hierarchical tree structure on the basis of hypernyms/hyponyms. The narrower term or concept (e.g. rose) is referred to as hyponym, whereas the broader term or concept is referred to as hypernym (e.g. flower). Hypernyms and hyponyms represent semantic relationships between synsets that are commonly assumed to be transitive. The semantics of concepts in the upper layers of the hierarchy are more generic, with lower similarity between them, whereas concepts at the lower layers or within the same layer are more concrete, with higher similarity [44].

One way to utilise WordNet to obtain the candidate values is to find a word's synsets that are sets of synonyms that are grouped together based on their meanings. We can also take advantage of WordNet's lexical association to find more specific words, for the original term (hyponyms). For example, suppose that we want to expand the word `religion`. We can look up the synsets for `religion` in WordNet and find other words that are synonyms, such as `faith`, `worship` and `sect`. We can also use WordNet's hyponyms to find more specific words, such as `Islam`, `Christianity` and `Buddhism`. Thus, the expanded words of `religion` can include `faith`, `worship`, `sect`, `Islam`, `Christianity` and `Buddhism`. Figure 3.7 presents a WordNet graph example for the term `religion` based on synsets and hyponyms.

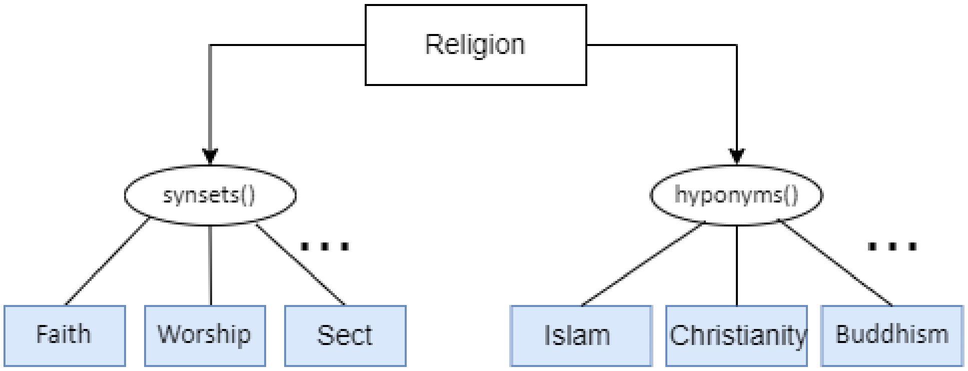


Figure 3.7: A part of the WordNet graph for "religion"

The pseudo-code given in Algorithm 3.1 is used to find the candidate values using

WordNet. Given a person characteristic (A, v), this algorithm aims to generate a set of candidate values $\{cv_1, cv_2, \ldots, cv_f\}$. Below is a detailed step-by-step breakdown of the algorithm:

1. If v is equal to $*$, assign the value of A to the variable `word`; otherwise, assign the value of v, to the variable `word`.
2. Use the synsets() function to look up the `word`. This function returns a set of synsets, which are groups of synonyms that refer to the same concept. For example, if the person characteristic is `(occupation,*)`, then the synonyms would be `job`, `career`, `profession`, etc.
3. For each synset obtained in the previous step, find the lemma. A lemma represents a specific sense of a particular word.
4. Find the hyponyms for each synset. Hyponyms are words that are more specific than the given word. This step helps to identify related words that have a narrower meaning. For example, hyponyms of `occupation` would be `teacher`, `farmer`, `tradesperson`, etc.
5. For each hyponym obtained, find the lemma.
6. Return the set of lemmas obtained from both the synsets and hyponyms. These lemmas represent the candidate values related to the `word`.

Algorithm 3.1 Candidate values generation by WordNet

```
Input: $(A, v)$                                  ▷ A person characteristic
Output: $\{cv_1, cv_2, \ldots, cv_f\}$           ▷ A set of candidate values from $A$ or $v$
$ifv == *$
  $word = A$
else:
  $word = v$
Look up word using synsets() function, (result is a set of synsets (synonyms)
  that all refer to the same concept)
Find lemma for each synset (each synset contains one or more lemmas, which
  represent a specific sense of a specific word)
Find hyponyms for each synset (result is words that are more specific than a
  given word)
Find lemma for each hyponym
Return lemma of synsets and hyponyms
```

We expand on one level hierarchy of WordNet, which consists of a synset, a set of synonyms with the same meaning, and their immediate hyponyms. Multi-layer hyponyms relationships can stretch too many words, thereby causing the basic keyword meanings to diverge. In other words, they can introduce noise into the results, thereby lowering matching accuracy. For example, `mozart specialists` are a fairly specific group of `teachers` as we move down the hierarchy in Figure 3.8. Several studies have suggested that using one level of WordNet produces the best results, including the research conducted by Gong et al. [23] (which studied ways to improve internet searches by using hypernyms and hyponyms in WordNet to broaden the query). They found that one level of hypernyms and hyponyms yields the best results. Fellbaum [18] also found that hyponymy works best when terms are close in the hierarchy, but not so effectively when terms are far apart in the hierarchy.

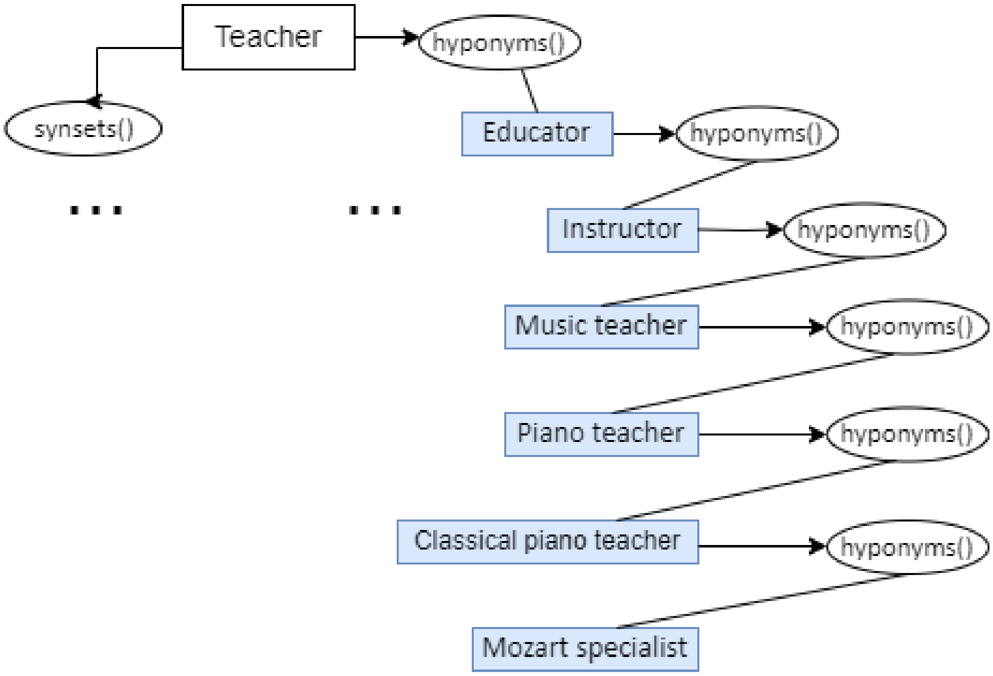


Figure 3.8: Example of multi-layer expansion on the word "teacher"

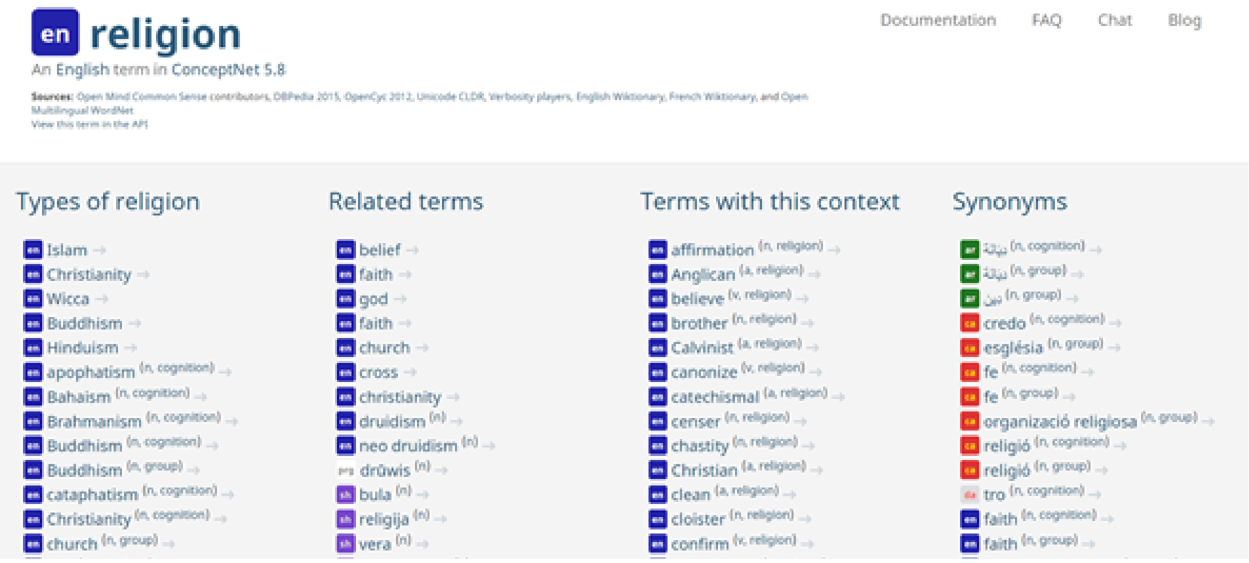
ConceptNet

Another popular knowledge base for semantic similarity is ConceptNet [49, 87, 86]. It uses common-sense relations like PartOf, UsedFor, and IsA. In the original semantic network, links between 300,000 nodes representing items constituted approximately 1.6 million statements of commonsense knowledge. However, subsequent editions have expanded and enhanced this. ConceptNet 5.5 [86] comprises over 21 million links between over 8 million nodes.

ConceptNet is a knowledge representation tool that enables the creation of a large semantic graph of general human knowledge and how it is represented in natural language. Not only does ConceptNet demonstrate how words are related by their lexical meanings, but it also reveals how they are related by common knowledge. For example, their understanding of “religion” extends beyond the properties that define it, such as “religion IsA belief system”, but also accidental facts such as those mentioned below:

- Types of religion (Islam, Christianity Buddhism, Hinduism, etc.)
- Related terms (belief, faith, God, cross, etc.)
- Synonyms (faith, organised religion, religious belief, etc.)

In our work, we use the relationship types of, related terms and synonyms to obtain the candidate values. Figure 3.9 presents an example of ConceptNet edges in a browsable interface that groups them according to their natural-language relationship.



en religion

An English term in ConceptNet 5.8

Sources: Open Mind Common Sense contributors, DBPedia 2015, OpenCyc 2012, Unicode CLDR, Verbosity players, English Wiktionary, French Wiktionary, and Open Multilingual WordNet

View this term in the API

Types of religion

- en Islam →
- en Christianity →
- en Wicca →
- en Buddhism →
- en Hinduism →
- en apophatism (n, cognition) →
- en Bahaism (n, cognition) →
- en Brahmanism (n, cognition) →
- en Buddhism (n, cognition) →
- en Buddhism (n, group) →
- en cataphatism (n, cognition) →
- en Christianity (n, cognition) →
- en church (n, group) →

Related terms

- en belief →
- en faith →
- en god →
- en faith →
- en church →
- en cross →
- en christianity →
- en druidism (n) →
- en neo druidism (n) →
- prg drūwis (n) →
- sh bula (n) →
- sh religija (n) →
- sh vera (n) →

Terms with this context

- en affirmation (n, religion) →
- en Anglican (a, religion) →
- en believe (v, religion) →
- en brother (n, religion) →
- en Calvinist (a, religion) →
- en canonize (v, religion) →
- en catechismal (a, religion) →
- en censer (n, religion) →
- en chastity (n, religion) →
- en Christian (a, religion) →
- en clean (a, religion) →
- en cloister (n, religion) →
- en confirm (v, religion) →

Synonyms

- ar ديانة (n, cognition) →
- ar ديانة (n, group) →
- ar دين (n, group) →
- ca credo (n, cognition) →
- ca església (n, group) →
- ca fe (n, cognition) →
- ca fe (n, group) →
- ca organizació religiosa (n, group) →
- ca religió (n, cognition) →
- ca religió (n, group) →
- da tro (n, cognition) →
- en faith (n, cognition) →
- en faith (n, group) →

Figure 3.9: Facts regarding “religion” from ConceptNet

Algorithm 3.2 is used to identify candidate values using ConceptNet. This algorithm takes as input a person characteristic represented by a pair (A, v). The goal is to generate a set of candidate values, represented as $\{cv_1, cv_2, \ldots, cv_f\}$. The algorithm operates in the following manner:

1. Check if the input is a variable (denoted by $*$), assign the value of A to the variable `word`; otherwise, assign the value of v to `word`.
2. Find the types of attributes, related terms, and synonyms for the `word`. For example, if the person characteristic is `(occupation,*)`, then the types of attribute would be `trade`, `farming`, `photography`, `accountancy`, etc; related terms would be `business`, `brewer`, `journalist`, etc; synonyms would be `work`, `profession`, etc.
3. For each synonym found in step 2, find the types of attributes and related terms. For example, our algorithm found that the words `work` and `profession` are synonyms for `occupation`. We find `profession` types of attribute (`lawyer`, `businessmen`, `nurse`, `pilot`, etc.) and related terms (`doctor`, `farmer`, `lecturer`, etc.).
4. Return all the words obtained from steps 2 and 3 as the set of candidate values.

WordNet and ConceptNet can both be used to obtain the candidate values. Both of them are general-purpose knowledgebases, thus can be applied to any attribute. However, WordNet and ConceptNet are different from one another in the following ways: WordNet is superior to ConceptNet in terms of quality and robustness due to the differences in their development processes (manually handcrafted vs automatically generated); and while WordNet focuses on formal taxonomies of words, ConceptNet focuses on a richer set of semantic relations between compound concepts [33].

Algorithm 3.2 Candidate values generation by ConceptNet

```
Input: $(A, v)$                                  ▷ A person Characteristic
Output: $\{cv_1, cv_2, \ldots, cv_f\}$           ▷ A set of candidate values from $A$ or $v$
1: $if v == *$
2:     $word = A$
3: else:
4:     $word = v$
5: Find types of attribute, related terms, and synonyms for $word$
6: For each synonym found in 1, find types of attributes and related terms
7: Return all words resulted from 5 and 6
```

3.3.2 Similarity Score Calculation

We now explain the second step of the LBAE stage. Having obtained the candidate values in the first step, we now determine whether a profile has the given attribute by employing the candidate values. To determine this, we measure the similarity between the terms in a profile and the candidate values using distributional approaches such as Word2vec. Distributional approaches utilise data from large corpora. The frequency with which two terms appear in the same context in a corpus is used to indicate the relationship between them [25, 28].

Note that the list of candidate values is constrained to the information obtained from the lexicon source, but people are likely to express similar information in diverse ways. Therefore, it becomes necessary to measure the distance between the candidate values and the terms in a profile in order to determine if the profiles actually includes the attribute. In the similarity score calculation step, we measure the degree to which the profile words and candidate values of the attribute are similar to determine whether the user who has this profile has this attribute.

In our work, we use word embeddings to measure the semantic similarity between the profile's words and the candidate values. The availability of many pre-trained models created by researchers that can be applied to a variety of domains, which makes them suitable for use in extracting any attribute from the hashtag community, is one benefit of the word embeddings approach.

Word Embedding (WE)

Word embedding is known as a distributional semantic model. It is a sort of word representation that enables machine learning algorithms to recognise words with related meanings. It is an approach for mapping words into vectors of real numbers. Word embedding is useful in a wide range of applications. One such use is predicting related and dissimilar words as well as the most frequently used keywords. Word2vec [61] and GloVe [77] are the most popular models for natural language processing to learn word vector by using a large corpus.

Word2vec learns word associations from large texts by using a neural network model. It is a strategy for capturing a huge number of precise syntactic and semantic word associations in natural language processing. Word2vec has two different frameworks: continuous bag of words (CBOW) and skip-gram. The CBOW model trains word vectors in a neural network by predicting a word based on its surrounding words within a specified context window size. In contrast, the skip-gram method predicts the target context words given a current word. The training architectures of these two models are depicted in Figure 3.10.

We used a pre-trained Word2vec model, which was trained on Google News. The model includes word vectors for three million words and phrases, which were trained on approximately 100 billion words. We did not train our own word embedding using Twitter profiles. This helps to establish the generality of our method: a word embedding trained on Twitter profiles could improve the accuracy of our work, but it

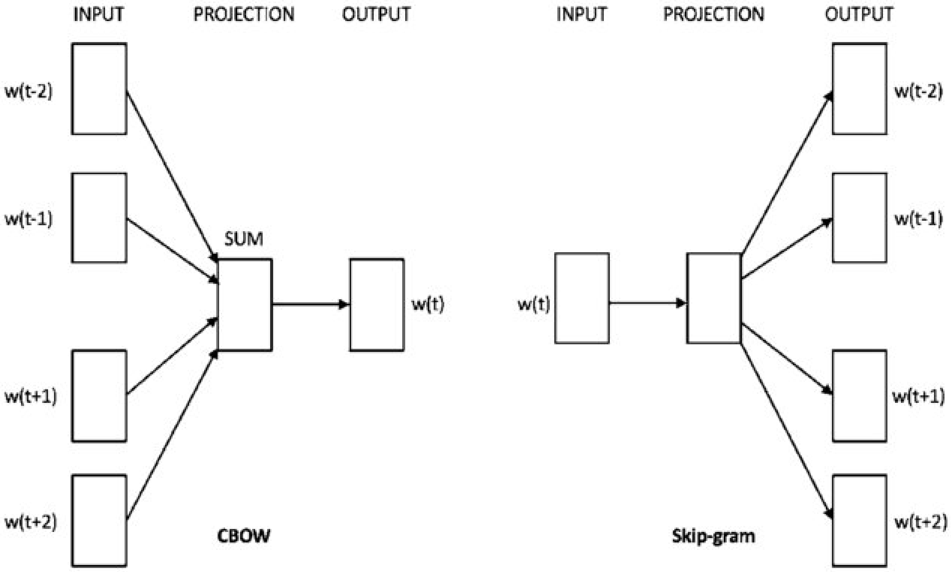


Figure 3.10: The continuous bag of words (CBOW) models versus skip-gram [61]

may not perform well on other platforms such as Facebook. The main disadvantage of using word embedding is the problem of out-of-vocabulary words. An embedding model cannot understand a word or know how to assign a vector to it if the word is not included in the training phase [7]. For example, in Figure 3.5, the similarity between the term `prolifer` in profile of U_1 and the expanded words cannot be calculated due to this fact (which `prolifer` is not included in the training phase).

Algorithm 3.3 provides a method to measure the similarity between a profile and different attribute values based on their candidate values. This algorithm takes as input candidate values list for all attributes, represented by $CV(A, v)$ and a profile Pj, and produces an output score list S that represents the similarity scores between the profile and all attribute values.

Algorithm 3.3 Similarity score calculation

```
Input: $CV\_(A, v)$    ▷ A candidate values for all attributes
       $P_j$    ▷ A profile
Output: $S$    ▷ A score list between a profile and all attribute values
For each row in $CV\_(A, v)$ do
    For each $cv_z$ in row do
        For each $t_k$ in $P_j$ do
            $Score(P_j, (A, v_i))$ = addToList(Word2vec.Sim($t_k$, $cv_z$))    ▷ $Score(P_j, (A, v_i))$ is a list has the similarity score between all terms in the profile and all candidate values of an attribute
    $S$ = addToList( $Max(Score(P_j, (A, v_i)))$)
return $S$
```

The algorithm utilizes the Word2vec [61] to calculate the similarity between each candidate value for an attribute and each term in the profile and adds it to a list, $Score(P_j, (A, v_i))$.

Then, it determines the maximum similarity score in the list $Score(P_j, (A, v_i))$ to represent the score between the profile and particular attribute value and stores this score in the list S. As an illustration of this, consider that we want to determine whether a given profile is `(Gender, male)`. Suppose that there are three candidate values for `(Gender, male)`: `man`, `dad` and `son`. As depicted in Table 3.3, as `dad` is more similar to the profile than `man` and `son`, the score between the profile and the attribute is, therefore, 0.62.

Table 3.3: Illustration of finding score between a profile and attribute `(Gender, male)`

		Candidate values		
		Man	Dad	Son
Profile terms	husband	0.16	0.48	0.47
	father	0.18	0.62	0.47
	grandfather	0.16	0.52	0.51
	patriot	0.13	0.24	0.21
	christian	0.07	0.13	0.14
	truth	0.11	0.12	0.12

Finally, the algorithm returns the score list S, which contains the scores between the profile and all values for the attribute in descending order.

3.3.3 The Extraction Process

Prior to providing a detailed explanation of the LBAE stage algorithm, we present a summary of the algorithm in Figure 3.11. The figure illustrates the process to determine attributes for one profile, P_1. In the first step, candidate values of the first attribute value (A, v_1) are found. In the second step, the similarity between each term in P_1 and each candidate value of (A, v_1) is calculated, and the maximum similarity value is taken to represent the similarity score between them. The first and second steps are repeated for other attribute values. This results in similarity scores between the attribute values and the profile. In the third step, the attribute/s of the profiles is decided or the profile is consider unknown. There are three different strategies to obtain this decision, as is evident from the algorithm.

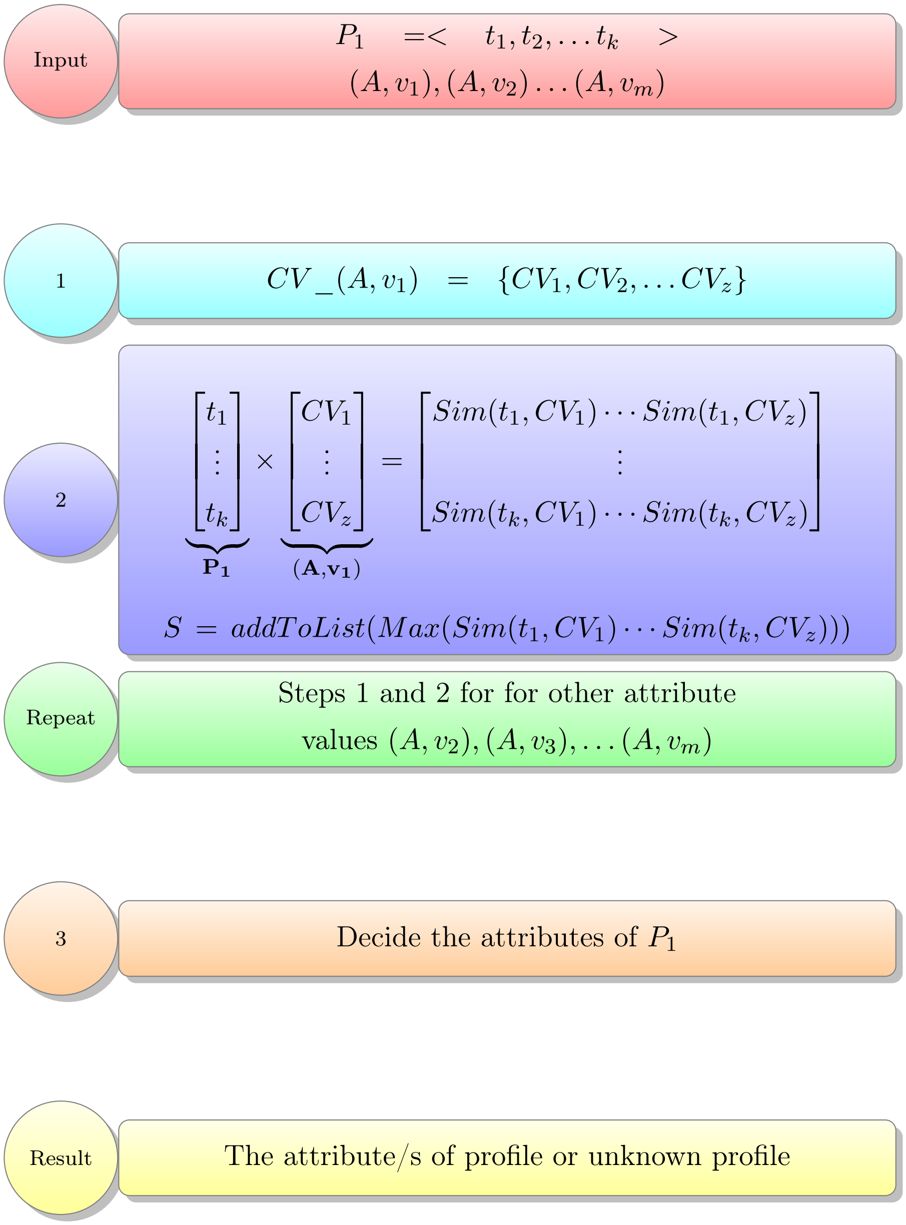

Input
$P_1 \quad =< \quad t_1, t_2, \ldots t_k \quad >$
$(A, v_1), (A, v_2) \ldots (A, v_m)$
1
$CV\_(A, v_1) \quad = \quad \{CV_1, CV_2, \ldots CV_z\}$
2
$\underbrace{\begin{bmatrix} t_1 \\ \vdots \\ t_k \end{bmatrix}}_{\mathbf{P_1}} \times \underbrace{\begin{bmatrix} CV_1 \\ \vdots \\ CV_z \end{bmatrix}}_{\mathbf{(A,v_1)}} = \begin{bmatrix} Sim(t_1, CV_1) \cdots Sim(t_1, CV_z) \\ \vdots \\ Sim(t_k, CV_1) \cdots Sim(t_k, CV_z) \end{bmatrix}$
$S \ = \ addToList(Max(Sim(t_1, CV_1) \cdots Sim(t_k, CV_z)))$
Repeat
Steps 1 and 2 for for other attribute
values $(A, v_2), (A, v_3), \ldots (A, v_m)$
3
Decide the attributes of P_1
Result
The attribute/s of profile or unknown profile

Figure 3.11: Summary of the extraction process

The LBAE stage is represented in Algorithm 3.4. This algorithm takes an input consisting of a set of person characteristics and all the profiles of community participants. While we implement our method within online communities, it does not require that the text exclusively originates from online communities. If we provide user profiles from a local tennis club, we can use the same approach to identify the characteristics of the individuals associated with this club. The algorithm produces three lists:

- The first list, denoted as C_p, contains profiles from which attributes are extracted.
- The second list, denoted as $LabelC_p$, contains the labels corresponding to the profiles in C_p.
- The third list, denoted as U_p, consists of profiles for which it is unknown whether they have attributes.

Algorithm 3.4 The LBAE stage

```
Input: $(A, v_1)$, $(A, v_2)$, . . . , $(A, v_m)$          ▷ Set of person characteristics
       $P_1$, $P_2$, . . . $P_n$                             ▷ Set of profiles
Output: $C_P$                                               ▷ Classified profiles
        $LabelC_p$                                          ▷ Labels of classified profiles
        $U_p$                                               ▷ Unknown profiles
 1: $C_P$ = [], $LabelC_p$ = [], $U_p$ = []
 2: For each $(A, v_i)$ do
 3:   $CV\_(A, v_i)$ = Find_candidate_values $(A, v_i)$ (Algorithm 3.1 or 3.2)
 4:   $CV\_(A, v)$ = addToList($CV\_(A, v_i)$) ▷ $CV\_(A, v)$ is a Two-dimensional list
    containing candidate values for all attributes
 5: For each $P_j$ do
 6:   $S$ = Similarity_Score_Calculation $(CV\_(A, v), P_j)$ (Algorithm 3.3)
 7:   input = Read input from the user      ▷ To determine which perspectives to
    follow, 1= Ranking-based, 2= Threshold-based
 8:   if input == 1:
 9:     Go to step 14
10:   if input == 2:
11:     for i in $S$:
12:       if i < $\alpha$
13:         deleteElement($S$, i)
14:   if $S$ list is empty
15:     $U_p$ = addToList($P_j$)                            ▷ The profile is unknown
16:   else:
17:     d = Read input from the user  ▷ To determine which strategies to follow.
    1= All-score strategy, 2= Highest-score strategy, 3. Eccentricity-based strategy
18:     labels = Determine_the_attribute $(d, S)$ (Algorithm 3.5)
19:     if len(labels) == 0:
20:       $U_p$ = addToList($P_j$)
21:     else:
22:       $C_P$ = addToList($P_j$),
23:       $LabelC_p$ = addToList($labels$)
```

The algorithm includes three steps.

Step 1 (lines 2–4): The algorithm finds the candidate values of each person characteristic using Algorithm 3.1 or 3.2.

Step 2 (lines 5 and 6): First, the algorithm calculates the similarity between each profile and each attribute value using Algorithm 3.3 and stores the score in the list S. Then, for each score in the list S, the algorithm evaluates whether the obtained score that represents a degree of similarity between a profile and an attribute value is considered to be acceptable to state that the profile has this attribute. There are generally two different perspectives on this. The algorithm reads input from the user to determine which perspective to follow, ranking-based or threshold-based (lines 7–13).

1. Relative measure (ranking-based)
   The score is acceptable without constraints (lines 8 and 9) because its value is defined having a high degree of similarity between the words in the profile and the candidate values of the attribute. However, the highest degree of similarity might still be a rather small value, which could lead to the identification of an incorrect attribute for the profile.

   Table 3.4 presents an illustration of this problem; the score (0.18) between profile and candidate values for `male` is the highest, but it is relatively low; hence, the profile is considered to have the `(Gender, male)` attribute. But this can lead to an incorrect attribute extraction.

Table 3.4: Illustration of ranking-based problem when attribute is `(Gender, male)`

		Candidate values		
		Man	Dad	Son
Profile terms	america	0.11	0.12	0.09
	thank	0.03	0.18	0.09
	new	0.02	0.06	0.03
	follower	0.14	0.16	0.14
	share	-0.001	0.04	-0.02
	informative	-0.01	0.06	-0.06

2. Absolute measure (threshold-based) A score is considered to be acceptable if it is above a specified threshold (α), and is included in the output score list (lines 10–13).

   For example, Table 3.5 illustrates the similarity of one profile to the attributes `male` and `female` using threshold-based perspective, with α set to 0.45. Both scores are listed because they are higher than the threshold. However, when a profile's similarity to the attributes `teacher`, `engineer` and `lawyer` is measured, the `lawyer's` score (0.4) is not listed in the score list because it falls below the threshold.

Table 3.5: Illustration of utilising a threshold-based perspective, with α set to 0.45

Attribute	$Score(P_j, A_i)$	Score list
A_1: Male	0.8	S = [(Male, 0.8), (Female, 0.5)]
A_2: Female	0.5	
A_1: Teacher	0.9	S = [(Teacher, 0.9), (Engineer, 0.8)]
A_2: Engineer	0.8	
A_3: Lawyer	0.4	

The specific use case and level of similarity required will determine the threshold value. A lower threshold value may be useful in certain applications to capture a larger range of similarities among words, while a higher threshold value may be desired in other applications to ensure that only highly similar words are actually considered similar.

Step 3 (Determine the attribute): The score list from the previous step may be empty or contain one or more scores. Multiple scores in the list indicate multiple acceptable attributes for the profile. Therefore, step 3 is to determine the attribute of the profile (lines 14–23). First, if the score list is empty, the profile is considered unknown (lines 14 and 15). This indicates that no attribute has a sufficient score to adequately represent the profile. If the score list is not empty (lines 16–23), there are three strategies to apply (see Algorithm 3.5). The algorithm takes two inputs: a list S and a value d that determine a strategy to be applied. The following are the three strategies that can be employed:

Algorithm 3.5 Determine the attribute

```
Input: $S = [score_1, score_2, \dots score_x]$    ▷ Score list of attributes $< A_1, A_2, \dots A_x >$
       $d$    ▷ The d value determines which strategy to follow
Output: labels    ▷ The label of profile can be one label, multiple labels or unknown
if $d == 1$    ▷ All-score strategy
  labels $= [A_1, A_2, \dots A_x]$
if $d == 2$    ▷ Highest-score strategy
  labels $= [A_i]$    ▷ $A_i$ has highest score with the profile
if $d == 3$    ▷ Eccentricity-based strategy
  if $(score_1 - score_2 > \gamma)$
    labels $= [A_i]$    ▷ $A_i$ has $score_1$
  else
    labels $= [\ ]$    ▷ The profile is considered unknown
return labels
```

1. The first strategy, called the all-score strategy, assumes that all the attributes with scores on the list will be used to characterise the profile. In certain circumstances, this strategy is helpful, but it is ineffective in others. Table 3.6 provides an example of two different scenarios. When a profile is first determined to be `male` or `female` by setting α to 0.45, the strategy returns both `male` and `female`. This information is useless because a person cannot be classified as `male` and `female` simultaneously. The strategy return may be helpful in the second scenario, when it is decided that a person is `a teacher`, `engineer` or `lawyer` using the same defined threshold. This may suggest that the profile is not indicating `lawyer`, but could indicate `teacher` or `engineer`.

2. The second strategy, referred to as the highest-score strategy, uses the attribute with the highest score to describe the profile. In the event of a tie, the attribute is selected at random. For example, in Table 3.6, the second strategy returns `male` in the first scenario and `teacher` in the second.

3. An eccentricity-based strategy is the third option. We utilised the concept of eccentricity in our research, which was originally introduced by [68]. In this strategy, if the difference between the highest and second-highest scores is greater than a certain threshold (γ), the attribute with the highest score is utilised to characterise the profile; otherwise, the profile is considered to be unknown. The use of this parameter ensures the similarity of a profile to one attribute is significantly greater than that of the rest. This measures how distinctive the classified attribute is in comparison to other attributes. For example, when the γ parameter is set to 0.2 in Table 3.6, the attribute of the first situation is `male`, as the difference between `male` and `female` is greater than 0.2 ($0.8-0.5 > 0.2$), while the attribute of the second situation is unknown because the difference between `teacher` and `engineer` is smaller than 0.2.

Table 3.6: Illustration of step 3 (Algorithm 3.5)

Attribute	Score list	First strategy (Return all)	Second strategy (Highest score based)	Third strategy (Eccentricity-based) $\gamma = 0.2$
A_1: Male, A_2: Female	Score[0.8,0.5]	Male Female	Male	Male
A_1: Teacher, A_2: engineer A_3: lawyer	Score[0.9,0.8]	Teacher engineer	Teacher	Unknown

3.4 Summary

In this chapter, we defined the attributes extraction problem. We also provided definitions of key concepts that are important to the understanding of this problem. We discussed two general approaches (search based and detection based) to extracting attributes from a community and explained the limitations associated with each approach. We also described our two-stage methodology for extracting attributes from hashtag communities. Then, we described the LBAE stage (first stage in our methodology) that involves finding candidate values and calculating similarity scores. The LBAE stage depends on two types of semantic similarity measures: knowledge-based similarity and distributional similarity. We used WordNet and ConceptNet as two knowledge-based approaches to obtain candidate values and used a technique from the distributional approaches, word embeddings, to determine whether a profile contains the attribute.

While the LBAE stage can classify profiles without labelled data and extract any attributes, there are cases in which certain profiles cannot be determined during this stage. However, we can learn from those profiles that the LBAE stage has managed to perform classification.

Chapter 4

Classification Enhanced Attribute Extraction

In this chapter, we describe the second stage in our proposed approach: the classification enhanced attribute extraction (CEAE). This stage utilizes the results from the previous LBAE stage as ground truth to train a classification model and then use the model to classify the profiles that were not determined in the LBAE stage. For example, the LBAE stage may have discovered that 30% of users are `female`, trained a model on these profiles, and used it to classify the remaining 70% users. If the CEAE stage is able to determine that an additional 20% of the profiles are `female`, then we will be able to extract this attribute for 50% of the profiles.

Classification is a machine learning tasks that is used to determine to which class or category a new observation belongs to based on a set of training data that includes observations whose classes are known. Our CEAE stage utilizes an iterative classification procedure and is designed to learn incrementally. During each iteration, the model updates its classification model and applies the revised model to unknown profiles until no further updates are necessary.

The remainder of this chapter is organised in the following manner: we present our basic CEAE stage and its improvement in Sections 4.1 and 4.2, respectively. The chapter is concluded in Section 4.3.

4.1 The Basic CEAE Stage

In the basic CEAE stage, we build a model using profiles whose attributes have already been extracted in the LBAE stage. It is considered basic because it does not account for classifier problems. The general interaction between the LBAE stage and the CEAE stage is illustrated in Figure 3.3 in Chapter 3.

The LBAE stage is rule-based, depending on candidate values derived from lexicon sources such as WordNet. When a candidate value closely matches a term in a profile, the profile is considered to possess that attribute. However, relying solely on the rules or candidate values is insufficient to extract the attribute from all profiles. This limitation arises from the limited candidate values available from a lexicon source and the varied ways in which users can express their attributes. The CEAE stage is learning-based and, thus, it learns from the results obtained in the LBAE stage.

The workings of our basic CEAE stage are presented in Figure 4.1. It consists of three main steps: feature extraction, model training, and classificaton of unknown profiles. A classifier is trained by examining a set of features extracted from the classified profiles resulting from the LBAE stage. Once trained, the model is subsequently used to classify unknown profiles.

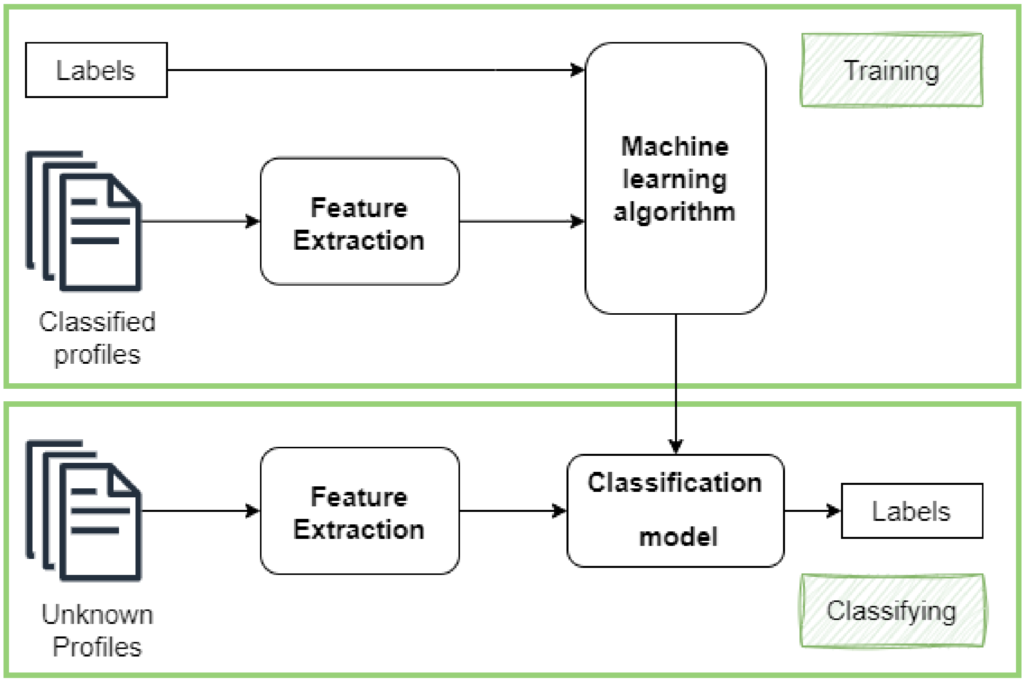


Figure 4.1: Block diagram of the CEAE stage

Algorithm 4.6 presents our basic CEAE stage. This algorithm takes a set of attributes $A_1, A_2, \ldots, A_m$, classified profiles (C_P) with their labels and unknown profiles (U_p) as inputs, and outputs the classification of U_p. A number of existing learning techniques (decision tree (DT), naive Bayes (NB) or support vector machine (SVM)) are used to build a model. When a classifier model is built, it can be used to classify all unknown profiles.

It is worth noting that in this work, we utilised some existing classification methods. While they influence our approach, they are used to support our proof of concept study. In other words, if better methods become available, our results would naturally improve. While refining semantic analysis (utilising tools like ChatGPT or BERT [16]) and optimising classifications through deep learning training can enhance overall performance, it is essential to note that we employ existing methodologies to demonstrate and validate the underlying concept.

Algorithm 4.6 The basic CEAE stage

```
Input: $A_1, A_2, \dots, A_m$        ▷ A set of attributes
       $C_P$                         ▷ Classified profiles
       $LabelC_p$                    ▷ Labels of classified profiles
       $U_p$                         ▷ Unknown profiles
Output: $C_P$                        ▷ Classified profiles
        $LabelC_p$                   ▷ Labels of classified profiles
Building a classifier using $C_P$ and $LabelC_p$
Using the classifier for $U_p$
Return the result of step 2
```

Consider a situation in which a profile must be assigned to one of several predefined attributes, but it does not clearly align with any of those attributes. In such cases, if the classifier was forced to assign an attribute to the profile, then it selects the closest or most similar attribute, even if it does not actually fit. This situation is commonly referred to as forced classification.

Forced classification occurs when an algorithm is required to assign an attribute to every profile, even when the profile does not have sufficient information for the attribute. This situation emphasises the importance of considering uncertainty, ambiguity, and the potential consequences of misclassification in classifier models. It is often preferable to refrain from assigning any attributes to a profile. To resolve the forced classification problem, we propose the following techniques:

1. Incorporate an additional class, such as “Not attributes” which represents profiles that cannot be confidently assigned to any existing attributes. By incorporating this class, the model can learn to classify profiles that do not correspond to the existing attributes into the “Not attributes” class, rather than compelling the classification of those profiles into the existing attributes. Thus, the addi-

tional class helps prevent incorrect classification that can occur when forcing a classification.

Consider a task of classifying community members into genders (`male` or `female`) using a model trained with members' profiles. Table 4.1 presents three profiles that cannot be assigned to either male or female categories and, hence, these profiles are considered the "Not attributes" class. A model can learn from these profiles as well as profiles classified as `male` or `female`. This enables the model to develop the ability to distinguish between profiles that belong to either the `male` or `female` category and those that are significantly different from them. Section 4.2.1 illustrates this technique.

Table 4.1: Example of "Not attributes" class

User (U)	Profile (P)
u_1	now comes the pain - wwg1wga
u_2	how much you know is determined by how much you are willing to disprove your previous beliefs.
u_3	planting seeds of truth and goodness for the great awakening!

2. Determine a threshold for uncertainty or confidence in classification: if the predicted probability or confidence score falls below a specified threshold, the profile can be considered uncertain and left unknown instead of forcing classification. This technique incorporates a level of uncertainty into the classification process. Setting a threshold enables us to clearly state the degree of certainty required to classify a given profile. It is possible to classify profiles with high confidence or use a lower threshold in return for a greater number of profiles

classified.

Consider a task of classifying community members into `gender` using a model trained from members' profiles. The model may not be able to produce a confident prediction when a profile description contains words or phrases that are commonly used to describe both `male` and `female`. Instead of a forced label, the model outputs a probability score for each `gender` to represent the degree of confidence in its prediction. If this scores falls below a specified threshold, the profile can be left unknown. This technique is demonstrated in Section 4.2.2.

In the next section, we introduce the improved CEAE stage which uses the additional class "Not attributes" and confidence scores techniques to address the forced classification problem.

4.2 The Improved CEAE Stage

In the improved CEAE stage, we propose techniques to solve the problem of forced classification. We introduce the "Not attributes" class and confidence threshold.

4.2.1 "Not attributes" Class

We propose the "Not attributes" class which includes profiles that extend beyond the predefined attributes. When a model is built using this class in addition to predefined attributes, it can distinguish between the profiles that have the attributes and those that do not. Thus, the model can classify the profiles to have one of the attributes or not have them instead of forcibly classifying profiles to have one of the attributes, which often leads to assigning them to inappropriate attributes.

For example, Table 4.2a presents the result when we search in a community for whether the participants can be categorised as `(Religion, Islam)`, `(Religion, Christianity)` or `(Religion, Hinduism)`. The results of the LBAE stage are presented in the second column, while the results of the CEAE stage are presented in the third column. The CEAE stage does not classify more profiles than LBAE for `(Religion, Islam)` and `(Religion, Hinduism)`. The results reveal that the majority of this community are categorised into `(Religion, Christianity)`. However, when we conduct a search within the same group to determine if they are in the category `(Religion, Hinduism)`, the outcomes (see second row in Table 4.2b) reveal that, completely unexpectedly, the community as a whole can be categorised as `(Religion, Hinduism)` (see the third column). This is because each desired attribute has a probability of being the label of the profile. For example, the probability of being `(Religion, Islam)` is 0.2, of being `(Religion, Christianity)` is 0.7 and of being `(Religion, Hinduism)` is 0.1 when we search in a profile for those three `religions`. Only searching for `(Religion, Hinduism)` yields a probability of 1. When a model is trained on a single attribute, it often generates a consistent output value of 1 for all profiles. This behavior is due to the model's ability to identify profiles belonging to the `(Religion, Hinduism)`, as it was trained on the premise that all profiles fall under this attribute. Therefore, the all members community in Table 4.2b (second row) is classified as `(Religion, Hinduism)`. The third row in Table 4.2b demonstrates how the findings improved and backed up the finding in Table 4.2a by using the "Not attributes" class.

In the LBAE stage, a profile is deemed to have the attribute or unknown based on the threshold α. If the similarity between the profile words and the attribute is greater than α, the profile is classified as having the attribute; otherwise, it is deemed unknown. When we build a classifier model utilising the results, this resulted in the

Table 4.2: Example of a forced classification problem

(a) When attributes are `(Religion, Islam)`, `(Religion, Christianity)` and `(Religion, Hinduism)`

	LBAE stage	CEAE stage
Support of attribute `(Religion, Islam)`	3.11%	3.11%
Support of attribute `(Religion, Christianity)`	12.95%	46.11%
Support of attribute `(Religion, Hinduism)`	1.04%	1.04%

(b) When attribute is `(Religion, Hinduism)`

	LBAE stage	CEAE stage
Forced classification problem	2.59%	100.00%
Adding “Not attribute” class	2.59%	2.59%

compulsory categorization of profiles that lack the attribute. This occurs due to the model's inability to discern between the patterns that distinguish the presence of targeted attributes from the absence of targeted attributes in the profiles. To address this issue, we modify the LBAE algorithm and use another threshold λ. We classify a profile as having an attribute if its similarity to the attribute's value is greater than α, and as not having the attribute when the similarity is less than λ. Otherwise, the similarity is between the two thresholds, and the profiles are classified as unknown. Table 4.3 presents the main difference between LBAE stage algorithm (Algorithm 3.4) and the modified LBAE stage algorithm.

Table 4.3: Difference between the LBAE stage and modified LBAE stage algorithms

LBAE stage algorithm	Modified LBAE algorithm
One threshold α	Two threshold α and λ
Two labels - $Similarity > \alpha$, has an attribute - $Similarity \leq \alpha$, unknown	Three labels - $Similarity > \alpha$, has an attribute - $Similarity < \lambda$, does not have an attribute - $\lambda \leq Similarity \leq \alpha$, unknown

Algorithm 4.7 presents the modified algorithm. In this section, we explain the changes made to the code discussed in Section 3.3.3 of Chapter 3. Similar to Algorithm 3.4, the modified algorithm retains all scores higher than the α threshold in list S, while scores less than or equal to the threshold are deleted. The remaining

scores are then sorted in decreasing order (lines 10–13). In contrast, when a score less than λ threshold, the counter value is increase (lines 14 and 15).

Then, if the score list is empty and the counter value equals the number of attributes, the label of the profile is “Not attributes” (lines 16–18). When the counter value is equal to the number of attributes, it implies that none of the attributes are similar to the profile. On the other hand, if the score list is empty and the counter value is equal to 0, the profile is unknown (lines 19–20). When neither of the two requirements is satisfied, the profile is classified using Algorithm 3.5 (lines 21–28), which covered in Section 3.3.3.

Algorithm 4.7 Modified LBAE stage

```
Input: $(A, v_1)$, $(A, v_2)$, ..., $(A, v_m)$                  ▷ Set of person characteristics
       $P_1$, $P_2$, ... $P_n$                                  ▷ Set of profiles
Output: $C_P$                                                   ▷ Classified profiles
        $LabelC_p$                                              ▷ Labels of classified profiles
        $U_p$                                                   ▷ Unknown profiles
1: $C_P$ = [], $LabelC_p$ = [], $U_p$ = []
2: For each $(A, v_i)$ do
3:    $CV\_(A, v_i)$ = Find_candidate_values $(A, v_i)$ (Algorithm 3.1 or 3.2)
4:    $CV\_(A, v)$ = addToList($CV\_(A, v_i)$) ▷ $CV\_(A, v)$ is a Two-dimensional list containing candidate values for all attributes
5: For each $P_j$ do
6:    $S$ = Similarity_Score_Calculation ($CV\_(A, v)$, $P_j$) (Algorithm 3.3)
7:    input = Read input from the user        ▷ To determine which perspectives to follow, 1= Ranking-based, 2= Threshold-based
8:    if input == 1:
9:       Go to step 19
10:   if input == 2:
11:      for i in $S$:
12:         if i $\leq \alpha$                ▷ The list $S$ after this step will have all scores $> \alpha$
13:            deleteElement($S$, i)
14:         if i $< \lambda$    ▷ When the attribute is dissimilar to the profile, the count is incremented
15:            count += 1
16:   if $S$ list is empty And count == m        ▷ m denotes the number of attribute values
17:      $C_P$ = addToList($P_j$),
18:      $LabelC_p$ = addToList(Not_attribute)
19:   if $S$ list is empty And count == 0
20:      $U_p$ = addToList($P_j$)                ▷ The profile is unknown
```

```
else:
    d = Read input from the user  ▷ To determine which strategies to follow. 1= All-score strategy, 2= Highest-score strategy, 3= Eccentricity-based strategy
    label = Determine_the_attribute $(d, S)$ (Algorithm 3.5)
    if len(labels) == 0:
        $U_p$ = addToList($P_j$)
    else:
        $C_P$ = addToList($P_j$),
        $LabelC_p$ = addToList($label$)
```

4.2.2 Confidence Score

A confidence score in probabilistic classification denotes the estimated likelihood or probability that a given profile belongs to a specific attribute. As an illustration, if a classification model is trained to categorise profiles of community participants as either `male` or `female`, and it can classify a particular profile as `male` with a confidence score of 0.85, which indicates that the algorithm believes that there is an 85% likelihood that the profile is actually a `male`. The minimal amount of confidence or probability needed for a predicted attribute to be accepted as a valid prediction can be determined by a certain confidence threshold (β). This confidence threshold can help address the problem of forced classification.

For example, Table 4.4 presents four profiles with their actual `gender` (second column) and the probability of a `female` or `male` classification (third and fourth columns), respectively. A conventional classification model would classify all profiles based on the highest probability (fifth column), thereby resulting in profiles 3 and 4 being incorrectly classified. Profiles 3 and 4 are not classified when the confidence threshold is set at 0.7 because their confidence scores are not sufficiently high (sixth column).

The interpretation of confidence scores can vary depending on the classification algorithm employed. For example, in naive Bayes, the confidence score is obtained through Bayesian probability estimation. After the model is trained on classified profiles, it calculates the posterior probability of each attribute, given the input features using Bayes' theorem [10]. The attribute with the highest posterior probability is selected as the predicted class, and the confidence score can be interpreted as the probability associated with that attribute. Another example is random forests, which are an ensemble learning method that combine multiple decision trees to make a classification. In random forests, predictions are made by aggregating the individual predictions from each decision tree in the ensemble. Each tree independently pre-

Table 4.4: Example illustrating the notion of a confidence score threshold β

Profile	Actual Gender	Predicted female probability	Predicted male probability	Predicted class using the basic CEAE stage	Predicted class using the improved CEAE stage β set to 0.7
1	Female	0.99	0.01	Female	Female
2	Male	0.03	0.97	Male	Male
3	Male	0.65	0.35	Female	Unknown
4	Female	0.44	0.56	Male	Unknown

dicts the attribute for a given profile, and the final prediction is determined through majority voting or averaging of the individual tree predictions.

We introduce a confidence threshold, β, in our classification. There are two requirements for a good confidence threshold. We would like to have as many profiles classified as possible and, simultaneously, the accuracy of the classification must be high. β should ideally maximise both the number of categorised profiles and accuracy, but this can be difficult to achieve in practice. Figure 4.2 presents the effect of the confidence threshold on classification. If we would to classify as many profiles as possible, then x is selected because it provides more categorised profiles, but the accuracy can be low. However, it may be necessary to maintain the highest level of accuracy in certain applications (i.e. obtaining accuracy y in Figure 4.2). Setting a high β will improve the accuracy of categorised profiles since only the most certain predictions will be accepted; however, certain profiles with the target attribute may be overlooked. In such scenarios, we contend that it is critical to think about how to strike a compromise between accuracy and categorised profiles. In other words, as z offers a good classified profiles and accuracy trade-off, we should seek to obtain z from Figure rather than x or y.

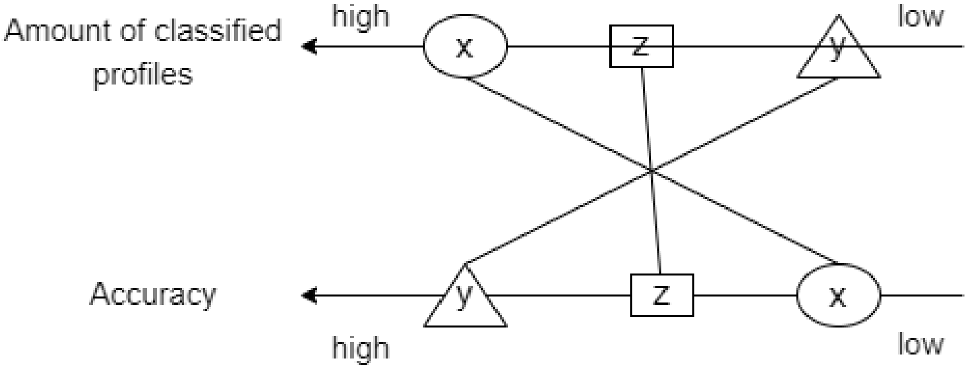


Figure 4.2: Classified profiles and accuracy trade-off

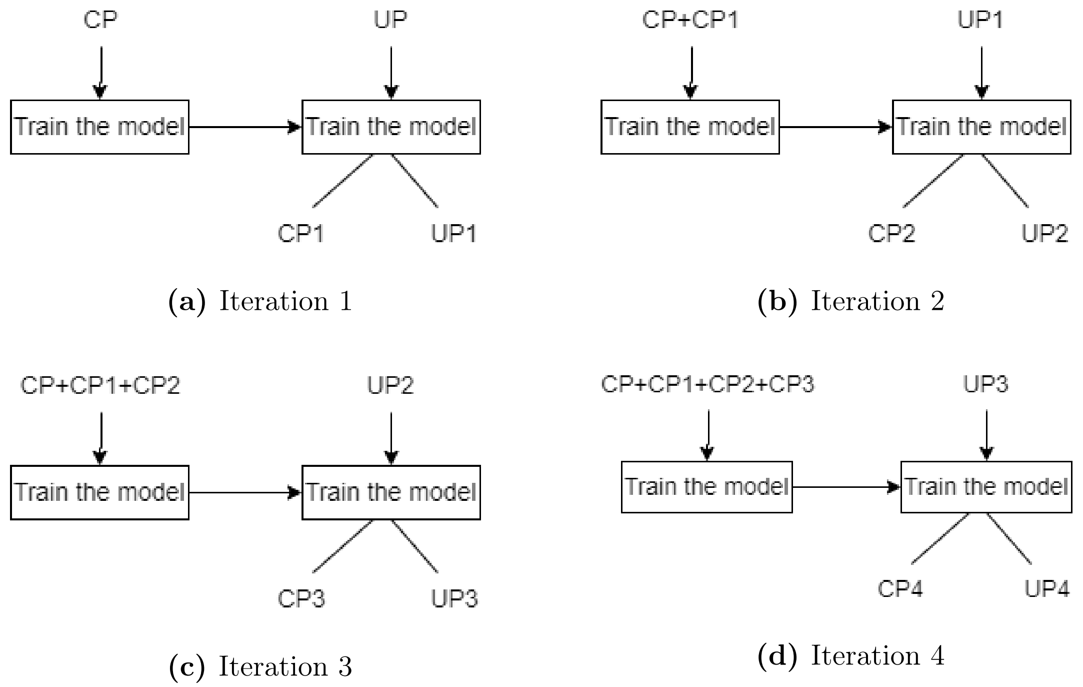
The improved CEAE stage improves the basic CEAE stage by adding the “Not attributes” class and confidence threshold. This resulted in profiles being classified with achieving a higher accuracy in classification. However, there are still unknown profiles. To help with these unknown cases, we developed an iterative classification method, as explained in the following section.

4.2.3 Iterative Learning

Iterative learning is a concept in machine learning that employs a repetitive process to progressively refine a model. It enables the model to continuously learn by integrating new classified profiles into the learning process; thus, learning more patterns improves its classification performance [55].

We apply two methods in iterative learning: traditional profile classification and complete profile classification. Traditional profile classification involves using classified profiles to classify unknown profiles. We train a model using classified profiles, and the model learns from these profiles and captures patterns and relationships between the input features and their corresponding attributes. Once trained, the model can be utilised to classify unknown profiles. However, in complete profile classification, we train the model on the classified profiles and use it to classify all the profiles, including the training classified profiles. This implies that the model provides classification for every profile in the dataset, including those it was originally trained on. The benefit of complete profile classification is that it enables us to evaluate the model's performance on the training classified profiles by examining how well the model classifies the attribute for the profile it was trained on.

In the LBAE stage and improved CEAE stage iterations, we have the classified profiles and unknown profiles. Let us refer to the entire profile set as P, the results of the LBAE stage as CP and UP, respectively. Let us also refer to the results of the first iteration of the improved CEAE stage as CP_1 and UP_1, respectively. The first four iterations of the traditional profile classification and complete profile classification are displayed in Figures 4.3 and 4.4. In both cases, the model is trained using CP in the first iteration; CP and CP_1 in the second iteration; CP, CP_1 and CP_2 in the third iteration; and CP, CP_1, CP_2, $\ldots CP_{n-1}$ in the nth iteration.



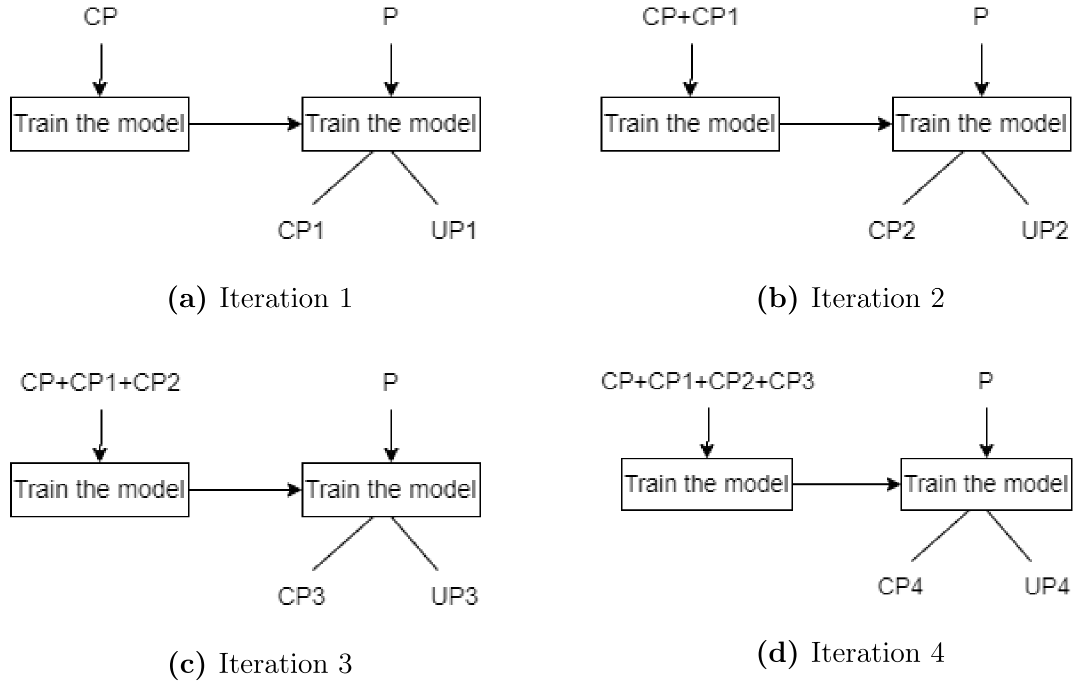
(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 4

Figure 4.3: Illustration of the improved CEAE stage iterations using traditional profile classification



(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 4

Figure 4.4: Illustration of the improved CEAE stage iterations using complete profile classification

Algorithm 4.8 reveals how the iterative CEAE works. Given a set of attributes $(A_1, A_2, \ldots, A_m)$, classified profiles (C_P) with their labels $(LabelC_p)$ and unknown profiles (U_p), a classifier is built using the results from the LBAE stage (C_P and $LabelC_p$) (line 2). The classifier is then used to determine the probability of the profiles in U_p (i.e. the probability that each unknown profile is deemed to have the attribute) (line 3). The profile is categorised by the attribute with the highest probability in the event that this probability exceeds β (lines 6 and 7). However, the profile remains unknown if this probability is less than or equal to β (lines 11 and 12). The steps from 2 to 12 are iterative until a condition is met. The condition can be accuracy or number of classified profiles. For example, users may require an accuracy of at least 80%. Since the objective is to classify as many profiles as possible, the method can continue to iterate as long as the accuracy is over 80%. In another case if a user wants to identify at least 70% of the participants in a community, then the method continues the classification until that condition is met. This may imply that we return a low accuracy for the attribute.

Algorithm 4.8 Improved CEAE stage

```
Input: $A_1, A_2, \ldots, A_m$                      ▷ A set of attributes
       $C_P$                                        ▷ Classified profiles
       $LabelC_p$                                   ▷ Labels of classified profiles
       $U_p$                                        ▷ Unknown profiles
Output: $C_P$                                       ▷ Classified profiles
        $LabelC_p$                                  ▷ Labels of classified profiles
        $U_p$                                       ▷ Unknown profiles
While Condition do
  Building a classifier model using $C_P$ with $LabelC_p$
  Probabilities = Using the classifier model to find the probability that each $u.p_1$
  can be predicted for each attribute.
  for each $u.p_j$
    for each $A_i$
      if probability of $A_i$ > probabilities of remain Attributes and $A_i > \beta$
        the profiles is $A_i$
        $C_P$ = addToList($p_j$)
        $LabelC_p$ = addToList($A_i$)
        $U_P$ = del($p_j$)
      else:
        the profiles is unknown
  end
```

One crucial aspect of iterative learning is the impact of earlier iterations cascades through subsequent iterations, thereby influencing the overall accuracy of the learning process. When each iteration captures patterns that are truly related to the desired attribute, thereby resulting in an overall improvement in accuracy. However, in certain cases, error patterns introduced in earlier iterations can persist and influence subsequent iterations, thereby leading to a decrease in accuracy.

Table 4.5 provides an example. Suppose we search for `male` in profiles. Simply to clarify the situation, we use a `father` term to identify a `male` in the LBAE stage. Participants whose profiles mentions `father` are classified as `male`, three profiles are classified (the first column), while the remainder are unknown. Now, a classifier model is trained using these profiles, which includes `Christian` as a classifying term. When the model is used to categorise unknown profiles, it labels the profiles that mention `Christian` as `male` and three more profiles are classified (the second column). These profiles mention `Christian`, as well as `conservative`, which was not mentioned in the prior round. The profiles that mentioned `conservative` are now classified, which implies that three more profiles are classified (the third column). We can show the impact of earlier iterations that results in categorising profiles that mention `Christian` and `conservative` as `male`.

Table 4.5: Example of how earlier iterations influence subsequent ones.

Classified profiles in the iteration 1	Classified profiles in the iteration 2	Classified profiles in the iteration 3
I'm God's servant, christian soldier, husband, father & all around family man. I seek the truth, desire justice, walk by faith, & express my heart with love	Independent Thinker, Independent is now Republican just to vote for our Trump! Constitutional Conservative, Christian, disgusted w/MSM.	retired USPS worker, conservative, Trump supporter, WWG1WWGA, MAGA, KAG, Trump 2020
Christian, Father, College Educated, #Trump supporter from day 1 #MAGA #Californians lets turn our state Red #Go TrumpPence2016	I'm a Veteran. That's Independent and is Conservative , and a Christian who believes that our country should get back to following the Constitution .	Conservative. Love Rush, Drudge, Lucianne, Breitbart, Michelle Malkin, Mark Levin, Mark Steyn, Greg Gutfeld, Tucker, Sara Carter Jesse Waters, Sharyl
CHRISTIAN, husband, father, grandfather, CONSERVATIVE University of Memphis 68. HUGE deplorable supporter of PRESIDENT DONALD J TRUMP	Independent Thinker, Independent is now Republican just to vote for our Trump! Constitutional Conservative, Christian ,disgusted w/MSM.	conservative, deplorable, MAGA ,married, animal lover, NO DM'S PLEASE, Go Trump

4.3 Summary

In this chapter, we discussed the second stage of our approach: the CEAE stage. This stage utilises the results from the LBAE stage to build classifiers for extracting attributes. First, we discussed the basic CEAE stage and discussed a problem it faces when classifying attributes, that is, the forced classification problem. Then, we suggested the improved CEAE stage by introducing a confidence threshold and adding a “Not attributes” class. Finally, we employed an iterative training process to our CEAE stage. By iteratively integrating and training on recently classified profiles, our model gains the ability to classify more profiles.

Chapter 5

Experiments and Results

In this chapter, we evaluate the effectiveness of the approach proposed in this thesis. We carried out two sets of experiments. First, we tested our LBAE and CEAE stages on benchmark datasets in terms of recall, precision, accuracy and F1 score. This was to observe how well our proposed methods work under various conditions; accordingly, we refer to this set of experiments as validation experiments. We then assessed the ability of our approach to extract attributes from a random set of hashtag communities. The chapter is organised as follows: In Section 5.1, we first describe the datasets used in our experiments and how they were prepared in Sections 5.1.1 and 5.1.2. We present the evaluation criteria utilised to assess the effectiveness of our work in Section 5.1.3. This is followed by a discussion of the validation experiments from Section 5.1.4 to Section 5.1.11. Finally, we describe the datasets used and discuss the results of the ability of our approach to extract attributes from a random set of hashtag communities in Section 5.2.

5.1 Results and Discussions of Validation Experiments

We began with the validation experiments, which evaluated our approach using benchmark datasets to establish its effectiveness. Once the level of accuracy was established, we validated our approach in extracting attributes from hashtag communities, assuming the same level of accuracy.

5.1.1 Datasets

The validation experiments were carried out on two Twitter datasets, as follows:

1. We tested our approach on one existing dataset (gender classifier data).[1] The dataset was used to assess how well the gender of a person could be classified. This data was the only publicly accessible source we were aware of for attribute extraction. This data categorised users into male, female or brand (non-individual). The classification was derived through crowdsourcing, and contributors were tasked with determining the gender of a user from their Twitter profiles. The dataset had 20,050 rows. We eliminated any user whose gender classification had a degree of confidence less than 1 to ensure that we were working with certain data. In addition, we removed the brand cases (non-individual), and only male and female profiles were used in our experiments. The filtered dataset contained 8421 users.

2. We also tested our approach on a Twitter dataset downloaded from Twitter using its streaming application programming interface (API). The data was

[1] https://data.world/crowdflower/gender-classifier-data

downloaded from 5 PM March 21, 2019 to 10 AM March 25, 2019, resulting in 60,041 rows containing user tweets and associated user profiles. After removing duplicate users and empty profiles, the dataset was reduced to 21,134 profiles. We then used human judgment to provide ground truth for these person characteristics (Religion, *), (Religion, Christianity), (Gender, Female) and (Gender, Male). Table 5.1 shows the number of profiles and the average number of words in each attribute.

Table 5.1: Dataset properties for each person's characteristics

Person Characteristic	Number of profiles	Average number of profiles' words
(Religion, *)	116	14.7
(Religion, christianity)	130	15.8
(Gender, female)	116	15.7
(Gender, male)	116	15.7

For each attribute, we ensured that our testing dataset comprised profiles with and without that attribute so as to assess our approach's ability to extract the attribute, whether it was present or absent. Each profile in the dataset was labelled with the attribute of interest by three people, and their consensus (majority) was used to label the profile.

5.1.2 Dataset Preparation

We preprocessed the user profiles as follows. We first removed web addresses, new line characters, hashtag symbols, single and double quotes, links and punctuation.

For hashtag terms, we deleted the # symbol while keeping the terms themselves. This was useful because when users add hashtags to their profiles, they can provide additional information about their affiliations, hobbies or the subjects they frequently tweet about. For example, including `#basketball` in their profiles may suggest their interest in basketball.

In the second step of dataset preparation, we tokenised the profiles using genism [80], removed stop words using NLTK [53], performed lemmatisation using spacy [32], and converted all letters to lowercase letters. Table 5.2a displays examples of profiles before and after pre-processing.

Finally, profiles were vectorised using CountVectorizer [75] for the CEAE stage. CountVectorizer is a popular method for transforming text input into a numerical representation that can be used by machine learning algorithms. Each distinct word or term in the profiles is represented as a feature in CountVectorizer. Table 5.2b shows an example of vectorised profiles in which each element in the vector corresponds to a specific feature.

Wang et al. [94] compared the term frequency-inverse document frequency (TF-IDF) and CountVectorizers in short text classification tasks. Whereas CountVectorizer merely counts the number of times each token appears in the text data, TF-IDF gives each token a weight, depending on how frequently it appears in the text data and throughout all the documents in the corpus. Wang et al. also used a variety of classifiers to compare tasks. Their research found that CountVectorizer is effective for short text classification tasks. Because TF-IDF relies on word frequencies in the document and across a collection of documents to determine word importance, TF-IDF faces challenges when dealing with short texts. The limited amount of content makes it difficult to obtain meaningful term frequencies and accurately represent documents.

User (U)	Profiles	After text pre-processing
U_1	author of “lost and found” and “love, dates and other nightmares”. co-host of podcast “reading in bed”. `https://twitter.com/reading_in_bed_`	author lost found love date nightmare co host podcast reading bed
U_2	#author of the #onehellofaromance series. like my page at `http://fb.com/jenniferfelton15`. #indieauthor #parnormal #romance #mystery #editing is my passion.	author onehellofaromance series like page indieauthor parnormal romance mystery editing passion

(a) Users’ profiles before and after the first and second steps of text pre-processing

Features	author, bed, co, dates, editing, found, host, indieauthor, like, lost, love, mystery, nightmares, onehellofaromance, page, parnormal, passion, podcast, reading, romance, series
U_1	[1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0]
U_2	[1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1]

(b) Users’ profiles after the third step of text pre-processing (Vectorisation)

Table 5.2: Dataset preparation

5.1.3 Evaluation Criteria

This section discusses the evaluation criteria used in the validation experiments. To determine how effectively our techniques can extract attributes, we first evaluated the LBAE stage, followed by the CEAE stage independently. We then assessed the overall effectiveness of our two-stage methodology.

When evaluating the LBAE stage and the CEAE stage independently, we used accuracy (a), precision (p) and recall (r) measures, which are commonly used to evaluate attribute extraction techniques [15, 85].

$$a = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$p = \frac{TP}{TP + FP} \tag{5.2}$$

$$r = \frac{TP}{TP + FN} \tag{5.3}$$

We also used the F-score to assess the overall quality of our method:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.4}$$

This offers an aggregation measure of recall and precision performance.

When evaluating the overall effectiveness of our two-stage methodology, we used the following formula:

$$Overall\ accuracy = (\frac{d1}{|D|} * a1) + (\frac{d2}{|D|} * a2) \qquad (5.5)$$

where $|D|$ is number of profiles, $\frac{d1}{|D|}$ is the fraction of profiles that can be classified by the LBAE stage, $a1$ is the LBAE stage accuracy measured by Equation 5.1, $\frac{d2}{|D|}$ is the fraction of profiles that can be classified by the CEAE stage and $a2$ is the CEAE stage accuracy measured by Equation 5.1. Thus, the overall accuracy of our approach is the weighted average of the two stages.

5.1.4 Evaluation of the LBAE Stage

In this section, we report the experiments on testing the effectiveness of the LBAE stage. In this set of experiments, we used the gender classifier dataset and the dataset we constructed ourselves with person characteristics (Religion, *), (Religion, Christianity), (Gender, Female) and (Gender, Male) manually annotated.

Experiment Setup

The following settings were used in our experiments:

- The candidate values were generated using one level of WordNet hyponyms and ConceptNet relationships. While multiple levels of hyponyms expansion are possible, we found that restricting to one level of expansion produced the best results, as explained in Section 3.3.1.

- In the score calculation step, the α values were varied to evaluate their impact on the effectiveness of our methods. We tested $\alpha = 0.65, 0.60, 0.55, 0.50, 0.45$ and 0.30.
- In the attribute selection step, we used the highest-score strategy to determine the profile's attribute, as explained previously in Section 3.3.3.

All the experiments were executed on a computer with 8GB main memory and an Intel(R) Core(TM) i5-8265U CPU @1.80 GHz running a Windows 10 operating system. Python 3.8.8 was used to implement the algorithms.

Results of Testing the LBAE Stage on the Gender Classifier Dataset

Table 5.3a shows the results of testing the LBAE stage on the gender classifier dataset. In this test, we used rank-based and threshold-based methods with various values for threshold α. In each value of α, the table displays the number of classified and unknown profiles.

Table 5.3: Results of testing the LBAE stage and the supervised learning methods on the gender classifier dataset

(a) Results of the LBAE stage

	Rank-based	Threshold-based					
		α=0.65	α=0.60	α=0.55	α=0.50	α=0.45	α=0.30
Recall	56.9%	86.2%	83.7%	85.5%	83.7%	82.4%	85.6%
Precision	50.2%	81.7%	77.6%	73.6%	68.7%	62.1%	53.3%
Accuracy	52.2%	83%	80.2%	76.7%	73%	66.8%	56.3%
F1 Score	53.3%	83.9%	80.5%	79.1%	75.5%	70.8%	65.7%
C_p	94.9%	6.3%	8.8%	12.1%	17.7%	24.1%	74.4%
U_p	5.1%	93.7%	91.2%	87.9%	82.3%	75.9%	25.6%

(b) Results of supervised learning methods

Supervised learning	DT	GNB	MNB	SVM
Recall	30.98%	37.13%	61.21%	51.25%
Precision	73.72%	62.61%	67.24%	64.81%
Accuracy	61.85%	59.48%	67.31%	63.51%
F1 Score	43.62%	46.61%	64.08%	57.24%
C_p	All	All	All	All
U_p	-	-	-	-

When the threshold-based method was used, accuracy grew monotonically with α, ranging from 56.3% to 83%. The recall appeared to be invariant to α, with values ranging from 82.4% to 86.2%. Precision increased monotonically with α, with values ranging from 53.5% to 81.7%. The F1 score, a weighted average of the previously described metrics, also seemed to grow monotonically with α from 65.7% to 83.9%. The proportion of classified profile entries decreased with α from 74.4% to 6.3%, whereas the percentage of unclassified profiles increased with α from 25.6% to 93.7%.

The rank-based method can classify a significantly higher number of profiles compared to the threshold-based method, but its accuracy is lower. The results of the rank-based method showed that it was approximately 50%, suggesting that the extraction results were comparable to random guessing. This result is due to accepting the similarity between profiles and candidate values without imposing any conditions, even when the similarities were low.

Table 5.3b shows a comparison of the classification of different supervised algorithm decision trees (DT), Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB) and a support vector machine (SVM) applied to the gender classifier data. Unlike our LBAE stage, these algorithms did not leave unclassified profiles. In terms of recall, MNB produced the best results, with a recall of 61.21%, followed by SVM (51.25%); the other two models had very low recall scores. This suggests that MNB and SVM perform better in capturing relevant profiles. The LBAE stage with the threshold-based method had a much higher recall than the supervised models reported. With regard to precision, the supervised models performed similarly to the LBAE stage using the threshold-based method; the precision scores ranged from 62.61% for GNB to 73.72% for DT. The accuracy of the LBAE stage was much greater than that reached by the supervised models, which ranged from 59.48% for GNB to only 67.31% for MNB. Supervised models can classify all profiles, whereas the LBAE stage can classify some of them but without the need for labelling.

Although the LBAE stage does not classify all profiles as supervised methods do, it is more accurate than supervised learning in some values of α. For instance, we were able to classify approximately 18% of the participants, with a 73% accuracy level when the threshold was established at 0.50 and roughly 24% of participants with a

66.8% accuracy level when the α was set to 0.45. We consider that although our LBAE stage does not classify every profile, it is a good achievement, given the absence of human labelling. Supervised learning is generally more effective in classifying all profiles, but its success relies heavily on human labelling. We can actually classify more profiles by decreasing the value of α. For instance, if we set α to 0.3, we could classify about 75% of the profiles, but the accuracy would be lower, at around 56%.

Results of the LBAE Stage on Multi-attribute Datasets

Tables 5.4, 5.5, 5.6 and 5.7 illustrate the results of testing our LBAE stage on the multi-attribute dataset (`(Religion, *)`, `(Religion, Christianity)`, `(Gender, female)` and `(Gender, male)`).

The recall in all datasets was highest for the rank-based method. As a rank-based method, it accepted any semantic similarity between candidate values and profile words, even if it was low; this resulted in capturing most of the profiles that had the attribute. However, the threshold-based method reached a similar value when α was set to 0.3. Slightly higher values of α (0.55) were enough to cause a large drop in recall in almost all datasets. When α was set to 0.65, it had the smallest recall in all datasets. This is because a higher α value means that a stronger semantic similarity between candidate values and profile words is needed; hence, the words have a greater degree of shared meaning. As the value of α rises, the threshold-based method captures fewer profiles that possess the attribute.

The precision of supervised learning models and the threshold-based method seemed to be superior to the rank-based method. The accuracy of the rank-based method compared to the threshold-based method was much lower, with average values around 30% compared to over 70% for the threshold-based method in the four datasets. The accuracy of the threshold-based method was comparable to that achieved by supervised learning techniques. For the F1 score, the threshold-based method with α smaller than 0.55 showed a better score than the rank-based method. The performance of the supervised learning models with regard to the F1 score was comparable to that of the threshold-based method (with α smaller than 0.55) across the datasets.

Our experiments showed that the LBAE stage was able to extract users' attributes from the profiles of community participants without human labelling of the data. However, many profiles remained unclassified. For instance, in Table 5.3a, the method could only identify an attribute for 18% of the members when α was set to 0.5.

Table 5.4: Results of testing the LBAE stage and the supervised learning methods on the `(Religion, *)` dataset

(a) Results of the LBAE stage

	Rank-based	Threshold-based					
		α=0.65	α=0.60	α=0.55	α=0.50	α=0.45	α=0.30
Recall	100%	5.4%	5.4%	16.2%	54%	81.1%	100%
Precision	32.7%	100%	100%	85.7%	95.2%	83.3%	38.1%
Accuracy	34.5%	69.8%	69.8%	72.4%	84.4%	88.8%	48.2%
F1 Score	49.3%	10.2%	10.2%	27.2%	68.9%	82.2%	55.2%
C_p	97.4%	1.7%	1.7%	6%	18.1%	32%	83.6%
U_p	2.6%	98.3%	98.3%	94%	81.9%	68%	16.4%

(b) Results of supervised learning methods

Supervised learning	DT	GNB	MNB	SVM
Recall	45.4%	86.3%	72.7%	40.9%
Precision	90.9%	42.2%	51.6%	100%
Accuracy	77.5%	50%	63.7%	77.5%
F1 Score	60.6%	56.7%	60.3%	58%
C_p	All	All	All	All
U_p	-	-	-	-

Table 5.5: Results of testing the LBAE stage and the supervised learning methods on the `(Religion, Christianity)` dataset

(a) Results of the LBAE stage

	Rank-based	Threshold-based					
		α=0.65	α=0.60	α=0.55	α=0.50	α=0.45	α=0.30
Recall	100%	2.1%	8.5%	10.6%	95.7%	95.7%	100%
Precision	37%	100%	100%	100%	100%	100%	58.7%
Accuracy	38.5%	64.6%	66.9%	67.6%	98.4%	98.4%	74.6%
F1 Score	54%	4.1%	15.6%	19.2%	97.8%	97.8%	74%
C_p	97.7%	0.8%	3.1%	3.8%	34.6%	34.6%	61.5%
U_p	2.3%	99.2%	96.9%	96.2%	65.4%	65.4%	38.5%

(b) Results of supervised learning methods

Supervised learning	DT	GNB	MNB	SVM
Recall	91.6%	91.6%	87.5%	62.5%
Precision	95.6%	44%	53.8%	93.7%
Accuracy	95.3%	53.8%	67.6%	84.6%
F1 Score	93.61%	59%	66.6%	75%
C_p	All	All	All	All
U_p	-	-	-	-

Table 5.6: Results of testing the LBAE stage and the supervised learning methods on the `(Gender, female)` dataset

(a) Results of the LBAE stage

	Rank-based	Threshold-based					
		α=0.65	α=0.60	α=0.55	α=0.50	α=0.45	α=0.30
Recall	100%	15.6%	50%	50%	87.5%	93.7%	100%
Precision	28.1%	55.5%	80%	69.5%	75.6%	60%	43.8%
Accuracy	29.3%	73.2%	82.7%	80.1%	88.7%	81%	64.6%
F1 Score	43.8%	24.3%	61%	58.1%	81.1%	73.1%	60.9%
C_p	98.3%	7.8%	17.2%	19.8%	31.9%	43.1%	62.9%
U_p	1.7%	92.2%	82.8%	80.2%	68.1%	56.9%	37.1%

(b) Results of supervised learning methods

Supervised learning	DT	GNB	MNB	SVM
Recall	52.9%	82.3%	88.2%	47%
Precision	90%	42.4%	55.5%	100%
Accuracy	84.4%	62.1%	75.8%	84.4%
F1 Score	66.6%	55.9%	68.1%	63.9%
C_p	All	All	All	All
U_p	-	-	-	-

Table 5.7: Results of testing the LBAE stage and the supervised learning methods on the `(Gender, male)` dataset

(a) Results of the LBAE stage

	Rank-based	Threshold-based					
		α=0.65	α=0.60	α=0.55	α=0.50	α=0.45	α=0.30
Recall	100%	23.08%	38.46%	100%	100%	100%	100%
Precision	11.4%	23.07%	27.78%	35.14%	34.21%	23.6%	12.38%
Accuracy	12.9%	82.76%	81.89%	79.31%	78.45%	63.8%	20.69%
F1 Score	20.5%	23.07%	32.25%	52%	50.98%	38.2%	22.03%
C_p	98.3%	11.2%	15.5%	31.9%	32.8%	47.4%	90.5%
U_p	1.7%	88.8%	84.5%	68.1%	67.2%	52.6%	9.5%

(b) Results of supervised learning methods

Supervised learning	DT	GNB	MNB	SVM
Recall	100%	25%	50%	100%
Precision	80%	11.1%	18.1%	80%
Accuracy	98.2%	81%	81%	98.2%
F1 Score	88.8%	15.3%	26.6%	88.8%
C_p	All	All	All	All
U_p	-	-	-	-

5.1.5 Effects of α

We investigated the impact of varying the α threshold on our LBAE stage. Increasing α resulted in an improvement in the LBAE stage's accuracy because a higher α value meant that a stronger semantic similarity between candidate values and profile words was needed; hence, the words had a greater degree of shared meaning. However, as the value of α rose, the number of classified profiles decreased.

It was observed that as α decreased, the LBAE stage captured fewer similar relationships, leading to higher recall. This meant that the model could identify a greater number of profiles that potentially contained the attribute as the threshold decreased. However, as the threshold decreased, precision tended to decrease as well, indicating that more profiles were misclassified. In other words, the model became less conservative in classifying profiles as having the attribute, which increased the likelihood of including incorrect classifications in the results.

5.1.6 Effects of Lexical Sources

We tested the effects of using different lexical sources in our LBAE stage. Table 5.8 shows the recall, precision, accuracy and F1 measures of the LBAE stage when using WordNet and ConceptNet as two different lexical sources. In this experiment, the value of α was fixed at 0.45 to observe the effect of these lexical sources on the results. We employed three ConceptNet relationships to find candidate values: types of, related terms and synonyms. We measured the use of types of relationship alone, the types of relationship and related terms and all relationships in the third, fourth and fifth columns, respectively. We observed that relying solely on types of relationship yielded worse results compared to incorporating additional relationships, except for the attribute `(Gender, male)`. This happened because by incorporating multiple relationships, the approach expanded the pool of potential candidate values, thereby enhancing the approach's ability to classify more profiles.

WordNet demonstrated a high average recall of 90%, which was comparable to the recall achieved by using the three relationships in ConceptNet (fifth column). The precision of WordNet was comparable to the precision achieved by using the types of relationship in ConceptNet (third column). Furthermore, WordNet exhibited superior accuracy in three out of four datasets compared to the three ConceptNet relationships, with an average accuracy value of 80%. When considering the F1 score, WordNet consistently outperformed the three ConceptNet relationships, with an average score of 80%. WordNet clearly demonstrated superiority over the three ConceptNet relationships with regard to precision, F1 score and the percentage of unclassified profiles. The accuracy and recall performances were similar.

We conducted this test to investigate whether our approach would be influenced by the use of different lexical sources. We observed that WordNet consistently outperformed ConceptNet across all attribute datasets except for `(Gender, male)` dataset. However, we acknowledge that utilising a more advanced lexical source, such as ChatGPT, could potentially improve our outcomes. We do not assert that our methodology succeeded solely due to our utilisation of WordNet or ConceptNet. Instead, our approach was formulated to capitalise on a lexical source as the initial stage. Considering the superior results obtained from WordNet, we continued to employ it in the subsequent experiments.

Table 5.8: Comparing measures in different lexical sources on multi-attribute datasets

Person characteristic	WordNet	ConceptNet (Types of)	ConceptNet (Types of and related terms)	ConceptNet (Types of, related terms and synonyms)
(Religion, *)	r: 81.1% p: 83.3% a: 88.8% f1: 82.2% C_p= 32% U_p= 68%	r: 56.8% p: 87.5% a: 83.6% f1: 68.9% C_p= 20.7% U_p= 79.3%	r: 89.2% p: 80.5% a: 89.7% f1: 84.6% C_p= 35.3% U_p= 64.7%	r: 89.2% p: 66% a: 81.9% f1: 75.9% C_p= 43.1% U_p= 56.9%
(Religion, christianity)	r: 95.7% p: 100% a: 98.4% f1: 97.8% C_p= 34.6% U_p= 65.4%	r: 29.8% p: 100% a: 74.6% f1: 45.9% C_p= 10.8% U_p= 89.2	r: 31.9% p: 93.8% a: 74.6% f1: 47.6% C_p= 12.3% U_p= 87.7%	r: 95.7% p: 81.8% a: 90.8% f1: 88.2% C_p= 42.3% U_p= 57.7
(Gender, female)	r: 93.7% p: 60% a: 81% f1: 73.1% C_p= 43.1% U_p= 56.9%	r: 56.2% p: 54.5% a: 75% f1: 55.4% C_p= 28.4% U_p= 71.6%	r: 100% p: 55.2% a: 77.6% f1: 71.1% C_p= 50% U_p= 50%	r: 100% p: 55.2% a: 77.6% f1: 71.1% C_p= 50% U_p= 50%
(Gender, male)	r: 100% p: 23.6% a: 63.8% f1: 38.2% C_p= 47.4% U_p= 52.6%	r: 100% p: 26% a: 68.1% f1: 41.3% C_p= 43.1% U_p= 56.9%	r: 100% p: 22.4% a: 61.2% f1: 36.6% C_p= 50% U_p= 50%	r: 100% p: 22.4% a: 61.2% f1: 36.6% C_p= 50% U_p= 50%

5.1.7 Evaluation of the CEAE Stage

In this set of experiments, we assessed our CEAE stage, which uses classifier models to extract attributes. To test iterative learning, the utilisation of substantial amounts of data was imperative. Therefore, we leveraged gender classifier data comprising approximately 8000 profiles.

Experiment Setup

In carrying out the experiments for the CEAE stage, the following settings were used:

- In the basic CEAE, we chose a number of well-known classifiers: decision tree, Naive Bayes and support vector machines.
- In the improved CEAE, we used the learning algorithms of the random forest classifier (RFC) and MNB. RFC and Naive Bayes are effective solutions for probabilistic classification issue.
- We experimented with varying confidence threshold values β to determine how they would affect the effectiveness of attribute extraction. We conducted tests on β values ranging from 0.7 to 0.85 in RFC, whereas in MNB, we were able to test up to 0.99. In RFC, when the threshold reached 0.85, the amount of classified data was relatively low, at approximately 0.3%. Thus, we decided to halt further testing. However, in MNB, when the threshold reached 0.85, the amount of classified data was significantly higher, at around 30%. Thus, we continued testing at higher β.
- In the iterative learning, α was set to 0.55. It was evident from the results of the improved CEAE that NB outperformed the other classifiers. NB was

therefore employed in the iterative learning. The confidence threshold β was set to 0.95.

Results of the Basic CEAE

Table 5.9 displays the results of the basic CEAE stage using various machine learning techniques with varying outcomes from the LBAE stage, employing different values of α. We found that the accuracy of Naive Bayes was better than that of other machine-learning algorithms. Naive Bayes performed well with CountVectorizer, which is commonly used for vectorisation. Additionally, the Naive Bayes model tends to excel when there is limited training data available [82, 69].

Although increasing the value of α in the LBAE stage led to higher accuracy, it appeared that in the basic CEAE, the accuracy was broadly similar for all α values since increasing the value of α in the LBAE stage resulted in a decrease in the number of classified profiles; this led to a reduction in the available training data. When the amount of training data decreased, the representation of the underlying patterns and variations in the data became less certain. On the other hand, using a lower value of α in the LBAE stage increased the number of classified profiles, thereby increasing the amount of training data for the basic CEAE. However, since the accuracy of the training data was lower in this scenario, it also led to a decrease in the accuracy of the basic CEAE.

Table 5.9: Results of the basic CEAE stage using gender classifier data

α	Basic CEAE (DT)	Basic CEAE (GNB)	Basic CEAE (MNB)	Basic CEAE (SVM)
0.65	r: 99.7% p: 47.1% a: 47% f1: 63.9%	r: 46.1% p: 54.7% a: 56.6% f1: 50%	r: 71.1% p: 50% a: 52.8% f1: 58.7%	r: 99.3% p: 47.2% a: 47.3% f1: 64%
0.60	r: 99.7% p: 47.2% a: 47.2% f1: 64.1%	r: 41.5% p: 54.4% a: 55.9% f1: 47.1%	r: 67.6% p: 50.6% a: 53.5% f1: 57.9%	r: 97.6% p: 47.4% a: 47.6% f1: 63.8%
0.55	r: 99.6% p: 46.7% a: 46.7% f1: 63.6%	r: 50.9% p: 53.4% a: 56.3% f1: 52.1%	r: 81.6% p: 49% a: 51.6% f1: 61.2%	r: 97.5% p: 46.9% a: 47.2% f1: 63.3%
0.50	r: 98.4% p: 46.8% a: 46.8% f1: 63.4%	r: 47.2% p: 52.8% a: 55.4% f1: 49.8%	r: 78.6% p: 48.4% a: 50.7% f1: 59.9%	r: 95.6% p: 46.8% a: 46.9% f1: 62.8%
0.45	r: 98.9% p: 46.9% a: 46.8% f1: 0.636	r: 47.8% p: 51.3% a: 54.2% f1: 0.495	r: 79.7% p: 47.3% a: 48.8% f1: 0.594	r: 79.7% p: 47.3% a: 48.8% f1: 0.594
0.30	r: 99.8% p: 42.9% a: 43% f1: 60%	r: 41.5% p: 9.7% a: 56.9% f1: 45.2%	r: 86.6% p: 42.1% a: 43.1% f1: 56.6%	r: 98.2% p: 42.6% a: 42.5% f1: 59.4%

Results of the Improved CEAE Stage

This section assesses the results of the improved CEAE stage, which addresses the forced classification problem.

For the experiments carried out for the improved CEAE stage, we used the RFC and MNB algorithms. Both methods made use of probability scores; thus, we used them to solve the forced classification issue. The RFC is an ensemble learning method that creates a number of decision trees and chooses the final result depending on the consensus of the individual trees. Each individual decision tree gives each class a probability score, and the result is determined by the majority of these values. The MNB utilises probability scores to make predictions. These probability scores play a crucial role in estimating the likelihood of an instance belonging to a specific class. The algorithm computes the conditional probability of each class based on the instance's features and identifies the class with the highest probability as the final prediction. This approach allows Naive Bayes to make informed and accurate predictions by leveraging probability estimation.

The results of utilising RFC with the numbers of estimators set at 100, 200, 300 and 400 are shown in Table 5.10. The number of estimators refers to the number of decision trees included in the ensemble. All of these results were obtained using α set to 0.55 and β set to various values. We also conducted MNB experiments (see Table 5.11). The tables demonstrate the outcomes of the basic CEAE and the improved CEAE with different levels of confidence threshold (β), where α is fixed at 0.55. We can clearly see that as the number of profiles classified increased, accuracy declined.

The basic CEAE stage is capable of classifying all profiles, but its accuracy is generally lower compared to the improved CEAE stage across the different values of β that we tested. In the basic CEAE stage using RFC, the accuracy remained around 49% across all estimator values used, whereas the accuracy in the improved CEAE stage varied within an average range of 48% to 75%. Similarly, the basic CEAE stage using MNB achieved an accuracy of 53%, while the improved CEAE stage exhibited an average accuracy ranging from 59% to 80%, with different values of β. However, as the accuracy of the improved CEAE stage increased, there was a corresponding increase in the amount of unclassified data.

Table 5.10: Results of the basic CEAE stage using RFC and the improved CEAE stage using RFC, when $\alpha = 0.55$

(a) The numbers of estimators = 100

	Basic CEAE	Improved CEAE $\beta =$							
		0.7	0.73	0.75	0.76	0.79	0.80	0.82	0.85
C_p	All	38.8%	13.3%	8.3%	6.4%	3.2%	1.8%	% 0.8%	0.3%
U_p	-	61.2%	86.7%	91.7%	93.6%	96.8%	98.2%	99.2%	99.7%
Accuracy	48.85%	49.36%	60.47%	63.19%	63.26%	59.75%	69.17%	72.58%	70.83%

(b) The numbers of estimators = 200

	Basic CEAE	Improved CEAE $\beta =$							
		0.7	0.73	0.75	0.76	0.79	0.80	0.82	0.85
C_p	All	46.5%	19.6%	10.9%	7.8%	2.4%	1.8%	0.9%	0.3%
U_p	-	53.5%	80.4%	89.1%	92.2%	97.6%	98.2%	99.1%	99.7%
Accuracy	48.64%	48.84%	55.90%	58.33%	61.07%	73.48%	74.80%	70.14%	75.0%

(c) The numbers of estimators = 300

	Basic CEAE	Improved CEAE $\beta =$							
		0.7	0.73	0.75	0.76	0.79	0.80	0.82	0.85
C_p	All	48.1%	19.9%	10.5%	7.6%	3.2%	1.8%	0.8%	0.3%
U_p	-	51.9%	80.1%	89.5%	92.4%	96.8%	98.2%	99.2%	99.7%
Accuracy	48.65%	48.84%	55.12%	60.75%	60.95%	63.17%	73.28%	73.92%	75%

(d) The numbers of estimators = 400

	Basic CEAE	Improved CEAE $\beta =$							
		0.7	0.73	0.75	0.76	0.79	0.80	0.82	0.85
C_p	All	44.8%	16.2%	8.4%	6.1%	2.1%	1.5%	0.6%	0.3%
U_p	-	55.2%	83.8%	91.6%	93.9%	97.9%	98.5%	99.4%	99.7%
Accuracy	48.60%	48.81%	56.61%	62.38%	64.89%	77.36%	72.97%	71.43%	84.21%

Table 5.11: Results of the basic CEAE stage using MNB and the improved CEAE stage using MNB, when $\alpha = 0.55$

	Basic CEAE	Improved CEAE $\beta =$						
		0.7	0.80	0.85	0.90	0.95	0.97	0.99
C_p	All	51.5%	35.6%	27.8%	20%	11.3%	7.8%	3.8%
U_p	-	48.5%	64.4%	72.2%	80%	88.7%	92.2%	96.2%
Accuracy	53.15%	59.30%	62.00%	63.83%	67.05%	72.14%	76.21%	80.21%

5.1.8 Effects of β

We investigated the impact of varying the β threshold on our improved CEAE stage. Increasing the value of β resulted in an improvement in the improved CEAE stage's accuracy since a high confidence threshold indicates a high probability that a given attribute is correct, thereby increasing accuracy. However, as the value of β increased, the amount of unknown profiles rose. Adjusting the β threshold helped strike a balance between accuracy and the classified profiles. By modifying the threshold, the user can influence the model's classification decisions and optimise its performance according to specific requirements.

5.1.9 Effects of Iteration

Table 5.12 shows the accuracy results of individual iterations in the improved CEAE stage. In this table, we applied the traditional profile classification method through iterative learning. The accuracy of each iteration was determined by evaluating the accuracy of the classified profiles found within that iteration. Accuracy was calculated using Equation 5.1. The first 8 iterations show how the process can classify more people. On the other hand, accuracy was generally reduced; the exception was that accuracy increased in iteration 8.

Table 5.12: Results of iterations in the improved CEAE stage using traditional profile classification

Improved CEAE stage	Accuracy	C_p	U_p
Iteration 1	77.22%	7%	93%
Iteration 2	66.05%	18%	82%
Iteration 3	63.60%	25.7%	74.3%
Iteration 4	65.43%	30%	70%
Iteration 5	62.05%	32.7%	67.3%
Iteration 6	53.66%	34.3%	65.7%
Iteration 7	58.59%	35.7%	64.3%
Iteration 8	65.38%	36.4%	63.6%
Iteration 9	55%	36.6%	63.4%

Suppose that we want to categorise 93% of the profiles left from iteration 1 in iteration 2. The model created in iteration 2 was able to classify more profiles and reduce the amount of unclassified profiles from 93% to 82%. Therefore, the classified profiles in iteration 2 must derive from the knowledge the model learned from the profiles that the iteration 1 model was able to classify.

In Table 5.13, we utilised the complete profile classification method for iterative learning. Each iteration involved the classification of all 8421 profiles. The accuracy of each iteration was evaluated by assessing the accuracy of the classified profiles within that specific iteration. Accuracy was calculated using Equation 5.1. Compared to the traditional profile classification method in Table 5.12, the complete profile classification method is more accurate and can classify more profiles. For example, in iteration 9, the accuracy was about 12% higher. The model was also able to decrease the amount of unknown profiles (53.5% compared to 63.4%). This resulted in higher accuracy because the model had already seen and learned from the training data, allowing it to make more accurate predictions on familiar profiles.

Table 5.13: Results of iterations in the improved CEAE stage using complete profile classification

Improved CEAE stage	Accuracy	C_p	U_p
Iteration 1	79.36%	14.8%	85.2%
Iteration 2	73.58%	26.9%	73.1%
Iteration 3	70.94%	34.7%	65.3%
Iteration 4	69.93%	39.5%	60.5%
Iteration 5	68.81%	42.5%	57.5%
Iteration 6	68.50%	44.3%	55.7%
Iteration 7	68.40%	45.2%	54.8%
Iteration 8	67.98%	36.1%	53.9%
Iteration 9	67.85%	36.6%	53.4%

The proposed approach utilises unsupervised techniques in the LBAE stage as a starting point, followed by the CEAE stage, to extract attributes without relying on labelled data. The findings demonstrated that both the LBAE and CEAE stages were effective in executing this task.

5.1.10 Evaluation of Overall Approach

This section presents the evaluation of the overall approach, which combined the LBAE stage with the basic or improved CEAE stage.

Table 5.14 shows the overall accuracy of our LBAE stage and the basic CEAE stage measured by Equation 5.5 using the gender classifier data. The accuracy of our approach varied across different models. In the decision tree and SVM, the accuracy ranged from 49.5% to 52.8%. For the GNB model, the accuracy ranged from 56% to 58%. In the MNB model, the accuracy ranged from 52% to 56%. By employing the basic CEAE stage, all remaining unknown profiles from the LBAE stage were classified.

We employed random classification as a baseline method to gain insights into the performance of our approach. Random classification classifies a profile as male or female randomly. We reported the average results over 10 runs. Our approach with the basic CEAE outperformed the baseline random classification with an accuracy of up to 58%, compared to the random classification's approximate 49.8%.

We compared our approach with different supervised learning algorithms, including DT, GNB, MNB and SVM. Although supervised learning has a higher accuracy rate than ours by about 9%, the advantage of our approach is that we do not require labelled data. Of course, we attempted to achieve the best level of accuracy possible, but we recognise that this is not always achievable. While there may be some accuracy differences between our approach and supervised learning at the moment, we believe it is acceptable and appropriate, given the possibility for future advancements.

Table 5.14: Comparing the overall accuracy of our approach (LBAE with basic CEAE) to random classification and supervised learning.
All profiles in our approach, random classification and supervised learning, are classified.

Random classification	49.83			
	DT	GNB	MNB	SVM
Supervised learning	62.17%	59.48%	67.31%	63.51%
Our approach when α = 0.65	49.41%	58.43%	54.86%	49.69%
Our approach when α = 0.60	49.99%	58.1%	55.91%	50.54%
Our approach when α = 0.55	50.22%	58.66%	54.53%	50.65%
Our approach when α = 0.50	51.52%	58.57%	54.71%	51.6%
Our approach when α = 0.45	51.6%	57.22%	53.12%	53.12%
Our approach when α = 0.30	52.84%	56.45%	52.43%	52.58%

Table 5.15 shows the overall accuracy measured by Equation 5.5 of our LBAE stage and the improved CEAE stage in the gender classifier data. In this experiment, we used MNB in the supervised learning and in our approach. While we found that when the confidence threshold was set to 0.85 or higher, our results were more accurate than supervised learning, we also noticed a rise in the number of unknown

profiles. MNB can classify every profile with 67.31% accuracy, while our approach achieved a classification rate of about 44%, with an accuracy of 66%. Notably, our approach achieved these results without the need for labelled data. When applying a confidence threshold in the improved CEAE stage, we followed a conservative approach by refraining from classifying instances when the model's confidence fell below the threshold. This cautious approach helped mitigate potential errors or misclassifications that could have arisen when the model's confidence was low.

Table 5.15: Comparing the overall accuracy of our approach (LBAE with improved CEAE), using MNB, and α= 0.55, to supervised learning, using MNB.

	Supervised learning	Improved CEAE β =						
	MNB	0.7	0.80	0.85	0.90	0.95	0.97	0.99
Accuracy	67.31%	62.97%	66.10%	68.09%	70.99%	74.66%	76.55%	77.50%
Amount of U_p	0	42.67%	56.63%	63.48%	70.32%	77.97%	81.06%	84.59%

5.1.11 Effects of the Unknown

We examined the reasons why we had unknown profiles after our two-stage methodology. We conducted a random manual examination of 60 profiles. Out of 60 profiles, 44 did not provide any relevant details regarding gender. These profiles lacked any form of descriptive information, making it impossible for the approach to extract gender from them. Thus, we successfully acquired them as unknown profiles. For example, Table 5.16 displays three profiles. Upon analysis, it was evident that determining gender as male or female was not possible due to the absence of clear indicators. In such cases, opting for an unknown classification is more appropriate than forcibly assigning these profiles a gender.

Table 5.16: Example of unknown profiles

User (U)	Profile (P)
u_1	i sing my own rhythm.
u_2	I'm the author of novels filled with family drama and romance.
u_3	you don't know me.

The remaining profiles (16 out of 60) comprised profiles written in languages other than English, empty profiles and profiles that contained information but were misclassified.

5.2 Results and Discussions of Extracting Attributes from Communities

In this test, we study how the developed method may be applied. That is, we assume that our method is able to extract an attribute from a given user profile with an average accuracy of 78%, as established in our validation test, and we are interested in determining the percentage of users in a hashtag group having that attribute, thereby shedding light on the characteristics of the group. For example, if we find that in a hashtag group, we have 5% females and 10% males, although each percentage is small, it helps us to see that this group is likely to have more males than females. This characteristic may help us understand this group. So, our objective in this test was not to measure the accuracy of our method in identifying users in a community with a particular attribute. Instead, we aimed to determine the percentage of users within a given hashtag community who possessed a specific attribute.

5.2.1 Datasets

In this experiment, we used Twitter's streaming API to collect hashtags. We searched for tweets containing hashtags and filtered non-English tweets. After collecting the data, we looked at hashtags with a significant number of unique individuals. Candidate hashtags had to have at least 50 users. Then, we removed hashtags if any of the following conditions were met: 1) the hashtag was a stop word, 2) it was a number, 3) it was not in English and 4) it was too brief or included only one or two letters. Since hashtags with too few characters are frequently used for general or unrelated purposes, they may be unable to effectively express key details about the topic. For instance, hashtags like #d, #in and #at may be too ambiguous to provide useful information about the posts or tweets with which they are used. To help the study focus on more valuable and relevant hashtags, these short hashtags were removed. The steps taken for data collection are summarised in Figure 5.1.

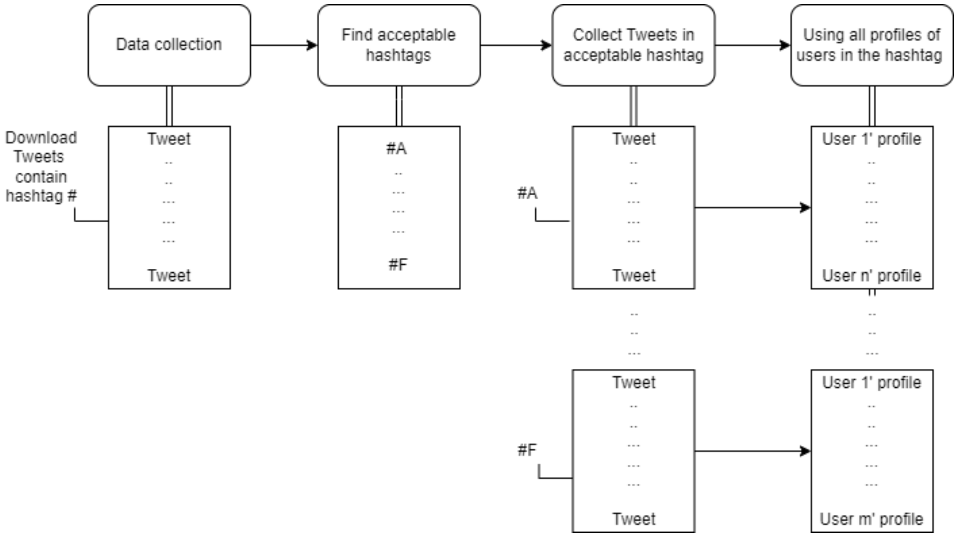


Figure 5.1: Data collection

After that, the user profiles were prepared for the experiments. The steps of data preparation have been discussed in Section 5.1.2.

5.2.2 Attribute Extraction for Hashtags

To verify our hypothesis, we selected hashtags that exhibited varying characteristics to ensure that our investigations into attribute extraction were conducted in a range of scenarios. We considered hashtags for which we could make assumptions about their attributes, as well as hashtags for which we could not. For example, when examining the hashtag #football, we might assume that it predominantly involves male participants, and we would like to see if our method would confirm this assumption. On the other hand, when it comes to the hashtag #coronavirus, we cannot make any definitive assumptions about the gender distribution of the participants, therefore it would be interesting to examine whether our approach suggests any particular gender distribution.

Tables 5.17 and 5.18 show the support for deriving several person characteristics over a range of hashtags. Table 5.17 presents the support results derived specifically from the LBAE stage, whereas Table 5.18 displays the support results obtained from both stages.

Table 5.17: The percentage of some person characteristics in some #hashtag communities using the LBAE stage,
person characteristics: 1. (Gender, male), 2. (Gender, female), 3. (Religion, *), 4. (Religion, Christianity), 5. (Political party, democrat), 6. (Political party, republican), 7. (Parent, *) and 8. (Occupation, teacher). The symbol N.P. refers to the number of participants.

Hashtag	N.P	1	2	3	4	5	6	7	8
#Brexit	5018	8.39	2.47	7.51	2.77	3.61	2.75	4.22	4.80
#coronavirus	24065	6.72	1.73	5.49	0.78	2.34	1.47	3.71	3.41
#Greatawakening	99	10.10	13.13	25.25	15.15	15.15	19.19	21.21	17.17
#Thegreatawakening	94	15.96	9.57	29.79	14.89	23.40	18.09	21.28	14.89
#Girlpower	912	9.43	4.17	5.04	14.79	1.86	1.75	7.68	5.37
#Cybersecurity	92	7.61	3.26	0	0	1.09	1.09	3.26	9.78
#artificialintelligence	71	9.86	2.82	0	0	0	0	4.23	5.63
#Humantrafficking	193	12.44	12.95	30.05	13.47	13.99	14.51	23.32	13.47
#rachelchandler	713	14.87	14.59	34.64	16.55	14.31	15.29	26.79	16.97
#qanon	962	14.24	11.12	29.42	14.03	16.63	17.36	21.93	14.14
#WWG1WGA	614	13.36	12.87	32.41	14.66	13.68	16.61	23.45	15.64
#Maga	365	15.34	11.78	31.51	14.79	16.16	18.36	23.84	14.52
#Muellerreport	100	27.00	24.00	29.00	11.00	33.00	19.00	24.00	19.00
#trump2020	116	26.72	24.14	31.03	12.07	27.59	15.52	23.28	14.66

Table 5.18: The percentage of some person characteristics in some #hashtag communities using the LBAE stage and the improved CEAE stage (RFC $\beta = 0.7$), person characteristics: 1. (Gender, male), 2. (Gender, female), 3. (Religion, *), 4. (Religion, Christianity), 5. (Political party, democrat), 6. (Political party, republican), 7. (Parent, *) and 8. (Occupation, teacher). The symbol N.P. refers to the number of participants.

Hashtag	N.P	1	2	3	4	5	6	7	8
#Brexit	5018	17.82	2.47	14.21	2.89	3.71	2.85	6.56	20.86
#coronavirus	24065	26.69	1.75	22.90	0.79	3.69	1.99	13.84	33.41
#Greatawakening	99	10.10	13.13	31.31	15.15	15.15	19.19	21.21	18.18
#Thegreatawakening	94	15.96	9.57	32.98	14.89	25.53	18.09	21.28	14.89
#Girlpower	912	13.38	4.17	5.04	15.89	1.86	1.75	7.79	5.37
#Cybersecurity	92	7.61	3.26	0	0	1.09	1.09	3.26	9.78
#artificialintelligence	71	9.86	2.82	0	0	0	0	4.23	5.63
#Humantrafficking	193	12.44	12.95	36.27	13.47	14.51	14.51	25.91	14.51
#rachelchandler	713	14.87	14.59	60.59	23.84	29.87	17.81	41.65	38.01
#qanon	962	14.24	11.12	49.79	21.52	33.78	19.44	35.55	28.27
#WWG1WGA	614	13.36	12.87	47.72	19.71	22.64	16.94	34.20	25.73
#Maga	365	15.34	11.78	47.95	15.89	26.03	18.90	32.05	23.56
#Muellerreport	100	38.00	27.00	36.00	11.00	33.00	19.00	28.00	21.00
#trump2020	116	36.21	25.00	43.10	12.07	27.59	15.52	24.14	14.66

We first analysed #artificialintelligence and #cybersecurity. There are relatively few women working in computer science—only 20%, according to [88]. In our analysis of these hashtags, males were more frequent than females, as expected. Specifically there were three times as many men as women in #artificialintelligence, and twice as many as in #cybersecurity (Table 5.18). Additionally, we noticed that individuals in both hashtags typically did not engage in talking about their political and religious beliefs, perhaps duo to the focus of the topics.

The hashtags #Greatawakening and #Thegreatawakening are about a number of periods of religious revival in American Christian history. We expected that both hashtags would have a majority of religious people, especially Christians. There were 99 participants in #Greatawakening and 94 in #Thegreatawakening. Eight users participated in both hashtags, whereas the remaining participants were different. Although these hashtags had almost entirely different participants, they yielded similar results using either the LBAE stage alone (Table 5.17) or both stages (Table 5.18). Moreover, in Table 5.19, we examined #Greatawakening with different religions, including Christianity, Islam and Hinduism and found that the results were in line with what we expected: the majority were Christians. In this table, we present the support results obtained using the following: LBAE stage alone, LBAE with basic CEAE stages and LBAE with improved CEAE stages.

In #Humantrafficking, the approach inferred that men and women would both support the hashtag equally. Both genders were expected to participate equally in this conversation. We discovered that 30% of participants were religious in general and that 50% of them identified as Christians.

#Muellerreport [97] is a hashtag about the Mueller Report, officially titled “Report on the Investigation into Russian Interference in the 2016 Presidential Election”. The #Muellerreport and #trump2020 hashtags had almost twice as many Democrats as Republicans. However, we observed that support for other topics was about the same across Democrats and Republicans. We can perhaps infer that Democrats were more interested in the discussion of #trump2020 and #Muellerreport, which is understandable.

The hashtag #Girlpower supports the strength, confidence, independence and empowerment of women. The outcome shows more awareness of male in women rights as the LBAE stage (Table 5.17) showed that males exceeded females by a small margin. When we incorporated the improved CEAE (Table 5.18), the gap between the number of male and female participants widened.

It is worth noting that, as our method was designed to extract any attribute from user profiles, we observed that in our experiments, it did not work equally well for all attributes. The approach performed well for some attributes, but not for others, because each attribute could have unique traits or patterns. For example, when we searched for "republican" and "democrat" as attributes, our extraction based on lexical sources such as Wordnet appeared to be limited because we only utilised their general English meaning, which is quite different than when they refer to their respective political parties; future work is needed to take this work forward. This requires an understanding of the context for the terms.

Table 5.19: The percentage of `(Religion, Islam)`, `(Religion, Christianity)` and `(Religion, Hinduism)` in the *#Greatawakening community*

Method	(Religion, Islam)	(Religion, Christianity)	(Religion, Hinduism)
LBAE stage	5.05	14.14	1.01
LBAE stage and basic CEAE (SVM)	6.06	23.23	1.01
LBAE stage and basic CEAE (DT)	13.13	19.19	1.01
LBAE stage and basic CEAE (RFC)	5.05	17.17	1.01
LBAE stage and basic CEAE (GNB)	14.14	48.48	1.01
LBAE stage and basic CEAE (MNB)	7.07	46.46	1.01
LBAE stage and improved CEAE (MNB $\beta = 0.9$)	5.05	14.14	1.01
LBAE stage and improved CEAE (RFC $\beta = 0.7$)	5.05	14.14	1.01

5.3 Summary

In this chapter, we evaluated our approach using existing and downloaded data. We studied the effectiveness of our approach by carrying out validation experiments. We tested the two stages of our approach separately in the validation experiments before evaluating the approach as a whole. We then examined our method's capacity to extract attributes from a random selection of hashtag communities.

The LBAE stage can achieve high accuracy in attribute extraction from profiles as α increases, but this improvement comes at the cost of reducing the number of classified profiles. Even though the basic CEAE stage can classify all profiles, the accuracy can only reach 56%. On the other hand, the improved CEAE stage can reach up to 80%, but with fewer classified profiles as accuracy increases. Iterative learning, on the other hand, helps to increase the number of classified profiles.

Our experimental results show that our approach can begin with unsupervised techniques (lexical source) and produce promising results. This approach can provide benefits for extracting different attributes without relying on labelled data.

Chapter 6

Conclusion

Many organisations frequently require insights into social media discussions, including an understanding of the characteristics of the participants in the community. In this thesis, we proposed a two-stage methodology to extract the attributes of online communities, specifically, Twitter communities. This chapter summarises and concludes our contributions and discusses possible directions for future study.

6.1 Research Summary

Understanding the characteristics of a social network community is important because it enables people, companies and organisations to better understand the wants and preferences of their target audience and develop more precise and successful marketing techniques. Several methods for extracting attributes from online social networks participants have been put forth in recent years; however, they rely on labelled data. Therefore, there is a need for an approach that does not depend on labelled data, making it applicable to any desired attributes.

In this thesis, we explore the application of semantic relationships and similarity to the automated extraction of desired attributes. This approach eliminates the necessity of manually labelling the initial dataset by leveraging lexical sources and semantic analysis as the first stage of our two-stage methodology. The outcomes of this initial stage serve as training data for constructing a classification model in the second stage.

We began our thesis by reviewing the limitations of current attribute extraction methods based on labelling data. Moreover, we present the research hypothesis, research questions, and contributions of this thesis in Chapter 1.

In Chapter 2, we reviewed the general concept of community. We then reviewed prior research in the area of attribute extraction in online communities. We concluded by discussing the research on semantic relationships that is relevant to our work.

In Chapter 3, we provided a thorough explanation of the first stage (LBAE) in our two-stage methodology, which comprises two main steps:

1. Candidate values generation is used to expand an attribute by a lexical source to improve the finding of an attribute in a profile. We used WordNet and ConceptNet to generate candidate values.
2. A similarity score calculation is used to measure how likely the profiles are to be similar to the candidate values. We used word embedding to measure the similarity between the profiles and the candidate values.

The LBAE stage has the ability to classify certain profiles without requiring explicit labelling, whereas supervised models can classify all profiles by relying on labelled data. Additionally, the accuracy achieved by the LBAE stage significantly surpasses that of the supervised models.

In Chapter 4, we described the second stage (CEAE) of our two-stage methodology, which builds a classifier model. We explained how iterative learning was developed and utilised to categorise additional community members who could not be categorised using the LBAE stage. We defined the term “confidence threshold” to refer to the relationship between accuracy and the quantity of classified profiles.

In Chapter 5, we presented the findings from the experiments. We provided information on the datasets and data preprocessing used in our investigation. The approach was tested to observe how well it worked and was assessed to show its ability to extract attributes from online communities.

Our two-stage methodology overcomes the limitations of previous methods by eliminating the need for labelled data and enabling the extraction of any desired attribute. In practice, semantic relationships and measures can be used to label the profiles. Once the profiles are labelled, they can be used to build a classification model to extract attributes. In our experiment, we successfully applied our methodology to a wide range of attributes. We achieved an average accuracy of 78% in attribute extraction. However, improvements in the accuracy of our methodology led to an increase in the number of unclassified profiles. We have effectively assessed the application of the developed method, determining the percentage of users within a specified hashtag community who exhibit a particular attribute. This analysis has yielded valuable insights into the characteristics of the group.

6.2 Future Work

Although this thesis has demonstrated the effectiveness of our attribute extraction approach, the work can be expanded in a number of ways:

- **Use member connections:**
  This thesis focuses on content-based communities rather than relying on connections among community members. Our analysis involved extracting attributes from profiles without utilising connections. However, connections can still play a supportive role in attribute extraction. For instance, if we observe that a person is connected to five individuals, and among them, four are supporters of a specific political party, we can reasonably assume that the person is more inclined to support that party.

- **Improve semantic measurement:**
  Twitter profiles contain abbreviations, spelling errors and slang, which can make it challenging to determine semantic equivalence. To generate suitable words for abbreviations, spelling errors and slang, we can approach it as a mapping or substitution problem. We need a dataset or a set of predefined rules that define the mappings between them and their corresponding full words. It is time-consuming to have a set of rules for each attribute, especially as our approach targets any attribute. The most suitable way is to use an automatic tool in the NLP that can generate suitable words for them.

- **Context consideration:**
  In this thesis, we have focused on measuring the similarity between an attribute and words in a profile. We cannot capture differences in context, such as distinguishing between “I am a wife” and “my wife” or “my student” and “I am a student”. It would be interesting to explore the incorporation of contextualised language models to enhance the understanding of profile contexts. A promi-

nent example of such a model is Bidirectional Encoder Representations from Transformers (BERT) [16], which has demonstrated state-of-the-art performance in various natural language processing tasks. Contextualised language models take into account the context in which a word or phrase appears. They capture the meaning and nuances of language by considering the surrounding words and sentences.

- **Data augmentation:**
  In our CEAE stage, the classification model's performance relies heavily on the results obtained in the LBAE stage. However, if the outcome is limited or imbalanced, it can have a significant impact on the effectiveness of the classification model. Data augmentation is a technique employed to tackle the challenges posed by limited and imbalanced training data in machine learning. Its objective is to augment the dataset artificially by generating additional examples through a variety of transformations or perturbations applied to existing data. One prevalent method of data augmentation is synonym replacement [103]. This approach involves substituting words in the text with their synonymous counterparts while maintaining the overall meaning of the sentence.

Bibliography

[1] Harshavardhan Achrekar et al. “Predicting flu trends using twitter data”. In: *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE. 2011, pp. 702–707.

[2] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. “Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 6. 1. 2012, pp. 387–390.

[3] Ashwaq Alsulami and Jianhua Shao. “Extracting Attributes for Twitter Hashtag Communities”. In: *International Journal of Humanities and Social Sciences* 16.3 (2022), pp. 171–178.

[4] Benjamin RC Amor et al. “Community detection and role identification in directed networks: understanding the twitter network of the care. data debate”. In: *Dynamic networks and cyber-security*. World Scientific, 2016, pp. 111–136.

[5] Jisun An and Ingmar Weber. “# greysanatomy vs.# yankees: Demographics and Hashtag Use on Twitter”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 10. 1. 2016, pp. 523–526.

[6] Oscar Araque, Ganggao Zhu, and Carlos A Iglesias. “A semantic similarity-based perspective of affect lexicons for sentiment analysis”. In: *Knowledge-Based Systems* 165 (2019), pp. 346–359.

[7] Dzmitry Bahdanau et al. "Learning to compute word embeddings on the fly". In: *arXiv preprint arXiv:1706.00286* (2017).

[8] Mohamed Bakillah, Ren-Yu Li, and Steve HL Liang. "Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan". In: *International Journal of Geographical Information Science* 29.2 (2015), pp. 258–279.

[9] Vuk Batanović and Dragan Bojić. "Using part-of-speech tags as deep-syntax indicators in determining short-text semantic similarity". In: *Computer Science and Information Systems* 12.1 (2015), pp. 1–31.

[10] Thomas Bayes. "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S". In: *Philosophical transactions of the Royal Society of London* 53 (1763), pp. 370–418.

[11] Nan Cao et al. "Socialhelix: visual analysis of sentiment divergence in social media". In: *Journal of Visualization* 18.2 (2015), pp. 221–235.

[12] Nina Cesare, Christan Grant, and Elaine O Nsoesie. "Detection of user demographics on social media: A review of methods and recommendations for best practices". In: *arXiv preprint arXiv:1702.01807* (2017).

[13] Abhijnan Chakraborty et al. "Who makes trends? understanding demographic biases in crowdsourced recommendations". In: *Eleventh International AAAI Conference on Web and Social Media*. 2017.

[14] Yahui Chen. "Convolutional neural network for sentence classification". MA thesis. University of Waterloo, 2015.

[15] Aron Culotta, Nirmal Kumar, and Jennifer Cutler. "Predicting the demographics of twitter users from website traffic data". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.

[16] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[17] Susan T Dumais et al. "Latent semantic analysis". In: *Annu. Rev. Inf. Sci. Technol.* 38.1 (2004), pp. 188–230.

[18] Christiane Fellbaum. "On the semantics of troponymy". In: *The semantics of relationships*. Springer, 2002, pp. 23–34.

[19] Evgeniy Gabrilovich, Shaul Markovitch, et al. "Computing semantic relatedness using Wikipedia-based explicit semantic analysis." In: *IJcAI*. Vol. 7. 2007, pp. 1606–1611.

[20] Walaa K Gad and Mohamed S Kamel. "New semantic similarity based model for text clustering using extended gloss overlaps". In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2009, pp. 663–677.

[21] Robert J Gaizauskas et al. "The University of Sheffield's TREC 2004 QA Experiments." In: *TREC*. 2004.

[22] Theodore Georgiou, Amr El Abbadi, and Xifeng Yan. "Extracting topics with focused communities for social content recommendation". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM. 2017, pp. 1432–1443.

[23] Zhiguo Gong, Maybin Muyeba, and Jingzhi Guo. "Business information query expansion through semantic network". In: *Enterprise Information Systems* 4.1 (2010), pp. 1–22.

[24] Jane Greenberg. "Automatic query expansion via lexical–semantic relationships". In: *Journal of the American Society for Information Science and Technology* 52.5 (2001), pp. 402–415.

[25] Kristina Gulordava and Marco Baroni. "A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus." In: *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*. 2011, pp. 67–71.

[26] Abram Handler. "An empirical study of semantic similarity in WordNet and Word2Vec". In: (2014).

[27] Sanda M Harabagiu et al. "Answer Mining by Combining Extraction Techniques with Abductive Reasoning." In: *TREC*. 2003, pp. 375–382.

[28] Sébastien Harispe et al. "Semantic similarity from natural language and ontology analysis". In: *Synthesis Lectures on Human Language Technologies* 8.1 (2015), pp. 1–254.

[29] Mariam Hassanein et al. "Predicting personality traits from social media using text semantics". In: *2018 13th International Conference on Computer Engineering and Systems (ICCES)*. IEEE. 2018, pp. 184–189.

[30] S Hiba and Y Keller. "Hierarchical attention-based age estimation and Bias estimation. arXiv 2021". In: *arXiv preprint arXiv:2103.09882* ().

[31] Angelos Hliaoutakis et al. "Information retrieval by semantic similarity". In: *International journal on semantic Web and information systems (IJSWIS)* 2.3 (2006), pp. 55–73.

[32] Matthew Honnibal and Ines Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". To appear. 2017.

[33] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. "Query expansion with conceptnet and wordnet: An intrinsic comparison". In: *Information Retrieval Technology: Third Asia Information Retrieval Symposium, AIRS 2006, Singapore, October 16-18, 2006. Proceedings 3*. Springer. 2006, pp. 1–13.

[34] Tianran Hu et al. “What the language you tweet says about your occupation”. In: *Tenth International AAAI Conference on Web and Social Media*. 2016.

[35] Xiaolei Huang et al. “Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition”. In: *arXiv preprint arXiv:2002.10361* (2020).

[36] Aminul Islam and Diana Inkpen. “Semantic text similarity using corpus-based word similarity and string similarity”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2.2 (2008), pp. 1–25.

[37] Hai Jin et al. “ComQA: Question answering over knowledge base via semantic matching”. In: *IEEE Access* 7 (2019), pp. 75235–75246.

[38] Xiaolin Jin, Shuwu Zhang, and Jie Liu. “Word semantic similarity calculation based on word2vec”. In: *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE. 2018, pp. 12–16.

[39] Tom Kenter and Maarten De Rijke. “Short text similarity with word embeddings”. In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 1411–1420.

[40] Kunal Khadilkar, Siddhivinayak Kulkarni, and Poojarani Bone. “Plagiarism Detection Using Semantic Knowledge Graphs”. In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE. 2018, pp. 1–6.

[41] Ari Z Klein, Arjun Magge, and Graciela Gonzalez-Hernandez. “ReportAGE: Automatically extracting the exact age of Twitter users based on self-reports in tweets”. In: *PloS one* 17.1 (2022), e0262087.

[42] Claudia Leacock and Martin Chodorow. “Combining local context and WordNet similarity for word sense identification”. In: *WordNet: An electronic lexical database* 49.2 (1998), pp. 265–283.

[43] Michael Lesk. “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone”. In: *Proceedings of the 5th annual international conference on Systems documentation*. 1986, pp. 24–26.

[44] Yuhua Li, Zuhair A Bandar, and David McLean. “An approach for measuring semantic similarity between words using multiple information sources”. In: *IEEE Transactions on knowledge and data engineering* 15.4 (2003), pp. 871–882.

[45] Yuhua Li et al. “Sentence similarity based on semantic nets and corpus statistics”. In: *IEEE transactions on knowledge and data engineering* 18.8 (2006), pp. 1138–1150.

[46] Kwan Hui Lim and Amitava Datta. “Finding twitter communities with common interests using following links of celebrities”. In: *Proceedings of the 3rd international workshop on Modeling social media*. ACM. 2012, pp. 25–32.

[47] Kwan Hui Lim and Amitava Datta. “Following the follower: detecting communities with common interests on twitter”. In: *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM. 2012, pp. 317–318.

[48] Dekang Lin et al. “An information-theoretic definition of similarity.” In: *Icml*. Vol. 98. 1998. 1998, pp. 296–304.

[49] Hugo Liu and Push Singh. “ConceptNet—a practical commonsense reasoning tool-kit”. In: *BT technology journal* 22.4 (2004), pp. 211–226.

[50] Xiaojian Liu, Yi Zhu, and Xindong Wu. “Joint user profiling with hierarchical attention networks”. In: *Frontiers of Computer Science* 17.3 (2023), p. 173608.

[51] Yaguang Liu and Lisa Singh. “Age inference using a hierarchical attention neural network”. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 3273–3277.

[52] Yaguang Liu, Lisa Singh, and Zeina Mneimneh. “A Comparative Analysis of Classic and Deep Learning Models for Inferring Gender and Age of Twitter Users [A Comparative Analysis of Classic and Deep Learning Models for Inferring Gender and Age of Twitter Users]”. In: *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications-DeLTA,* 2021.

[53] Edward Loper and Steven Bird. “Nltk: The natural language toolkit”. In: *arXiv preprint cs/0205028* (2002).

[54] Puneet Singh Ludu. “Inferring gender of a Twitter user using celebrities it follows”. In: *arXiv preprint arXiv:1405.6667* (2014).

[55] Yong Luo et al. “An appraisal of incremental learning methods”. In: *Entropy* 22.11 (2020), p. 1190.

[56] John Lyons and Lyons John. *Linguistic semantics: An introduction.* Cambridge University Press, 1995.

[57] Sunghwan Mac Kim et al. “Demographic inference on twitter using recursive neural networks”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2017, pp. 471–477.

[58] Johnnatan Messias, Pantelis Vikatos, and Fabrıcio Benevenuto. “White, man, and highly followed: Gender and race inequalities in Twitter”. In: *Proceedings of the International Conference on Web Intelligence*. ACM. 2017, pp. 266–274.

[59] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. “Corpus-based and knowledge-based measures of text semantic similarity”. In: *Aaai.* Vol. 6. 2006. 2006, pp. 775–780.

[60] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[61] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[62] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[63] Alan Mislove et al. "Measurement and analysis of online social networks". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement.* 2007, pp. 29–42.

[64] Alan Mislove et al. "Understanding the demographics of Twitter users". In: *Proceedings of the International AAAI Conference on Web and Social Media.* Vol. 5. 1. 2011, pp. 554–557.

[65] Alan Mislove et al. "You are who you know: inferring user profiles in online social networks". In: *Proceedings of the third ACM international conference on Web search and data mining.* 2010, pp. 251–260.

[66] Muhidin Mohamed and Mourad Oussalah. "A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics". In: *Language Resources and Evaluation* 54.2 (2020), pp. 457–485.

[67] Antonio A Morgan-Lopez et al. "Predicting age groups of Twitter users based on language and metadata features". In: *PloS one* 12.8 (2017), e0183537.

[68] Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets". In: *2008 IEEE Symposium on Security and Privacy (sp 2008).* IEEE. 2008, pp. 111–125.

[69] Kamal Nigam et al. "Text classification from labeled and unlabeled documents using EM". In: *Machine learning* 39 (2000), pp. 103–134.

[70] Nishant Nikhil and Muktabh Mayank Srivastava. "Content based document recommender using deep learning". In: *2017 International Conference on Inventive Computing and Informatics (ICICI).* IEEE. 2017, pp. 486–489.

[71] Jiaqi Pan et al. “Twitter Homophily: Network Based Prediction of User’s Occupation”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2633–2638.

[72] Deepa Paranjpe, Ganesh Ramakrishnan, and Sumana Srinivasan. “Passage Scoring for Question Answering via Bayesian Inference on Lexical Relations.” In: *TREC*. Citeseer. 2003, pp. 305–210.

[73] Sangameshwar Patil and Balaraman Ravindran. “Predicting software defect type using concept-based classification”. In: *Empirical Software Engineering* 25.2 (2020), pp. 1341–1378.

[74] Atish Pawar and Vijay Mago. “Challenging the boundaries of unsupervised learning for semantic similarity”. In: *IEEE Access* 7 (2019), pp. 16291–16308.

[75] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[76] Marco Pennacchiotti and Ana-Maria Popescu. “A machine learning approach to twitter user classification”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 5. 1. 2011, pp. 281–288.

[77] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[78] Dragomir Radev, Eduard Hovy, and Kathleen McKeown. “Introduction to the special issue on summarization”. In: *Computational linguistics* 28.4 (2002), pp. 399–408.

[79] Daniel Ramage, Anna N Rafferty, and Christopher D Manning. “Random walks for text semantic similarity”. In: *Proceedings of the 2009 workshop on graph-based methods for natural language processing (TextGraphs-4)*. 2009, pp. 23–31.

[80] Radim Rehurek and Petr Sojka. “Gensim–python framework for vector space modelling”. In: *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3.2 (2011).

[81] Qiufeng Ren et al. “Resource recommendation algorithm based on text semantics and sentiment analysis”. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, pp. 363–368.

[82] Irina Rish et al. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[83] H Andrew Schwartz et al. “Personality, gender, and age in the language of social media: The open-vocabulary approach”. In: *PloS one* 8.9 (2013), e73791.

[84] Taihua Shao et al. “Transformer-based neural network for answer selection in question answering”. In: *IEEE Access* 7 (2019), pp. 26146–26156.

[85] Luke Sloan et al. “Who tweets? Deriving the demographic characteristics of age, occupation and social class from Twitter user meta-data”. In: *PloS one* 10.3 (2015), e0115545.

[86] Robyn Speer, Joshua Chin, and Catherine Havasi. “Conceptnet 5.5: An open multilingual graph of general knowledge”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[87] Robyn Speer, Catherine Havasi, et al. “Representing general relational knowledge in conceptnet 5.” In: *LREC*. Vol. 2012. 2012, pp. 3679–86.

[88] ComputerScience.org Staff. *Women in Computer Science: Getting Involved in STEM*. [accessed 1-October-2022]. 2022. URL: https://www.computerscience.org/resources/women-in-computer-science/.

[89] Marco Vicente, Fernando Batista, and Joao P Carvalho. “Gender detection of Twitter users based on multiple information sources”. In: *Interactions Between Computational Intelligence and Mathematics Part 2*. Springer, 2019, pp. 39–54.

[90] Sarah Vieweg et al. “Microblogging during two natural hazards events: what twitter may contribute to situational awareness”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 1079–1088.

[91] Prashanth Vijayaraghavan, Soroush Vosoughi, and Deb Roy. “Twitter demographic classification using deep multi-modal multi-task learning”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2017, pp. 478–483.

[92] Svitlana Volkova, Yoram Bachrach, and Benjamin Van Durme. “Mining user interests to predict perceived psycho-demographic traits on twitter”. In: *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE. 2016, pp. 36–43.

[93] Wafa Wali, Bilel Gargouri, and Abdelmajid Ben Hamadou. “Enhancing the sentence similarity measure by semantic and syntactico-semantic knowledge”. In: *Vietnam Journal of Computer Science* 4.1 (2017), pp. 51–60.

[94] Ye Wang et al. “Comparisons and selections of features and classifiers for short text classification”. In: *Iop conference series: Materials science and engineering*. Vol. 261. 1. IOP Publishing. 2017, p. 012018.

[95] Zijian Wang et al. “Demographic inference and representative population estimates from multilingual social media data”. In: *The world wide web conference*. 2019, pp. 2056–2067.

[96] Michael White et al. “Multidocument summarization via information extraction”. In: *Proceedings of the first international conference on Human language technology research*. 2001.

[97] wikipedia. *Mueller report.* [accessed 1-Jan-2023]. 2023. URL: `https://en.wikipedia.org/wiki/Mueller_report`.

[98] Zach Wood-Doughty et al. "Predicting Twitter User Demographics from Names Alone". In: *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media.* 2018, pp. 105–111.

[99] Zach Wood-Doughty et al. "Using noisy self-reports to predict twitter user demographics". In: *arXiv preprint arXiv:2005.00635* (2020).

[100] Zhibiao Wu and Martha Palmer. "Verb semantics and lexical selection". In: *arXiv preprint cmp-lg/9406033* (1994).

[101] Take Yo and Kazutoshi Sasahara. "Inference of personal attributes from tweets using machine learning". In: *2017 IEEE International Conference on Big Data (Big Data).* IEEE. 2017, pp. 3168–3174.

[102] Emilio Zagheni et al. "Inferring international and internal migration patterns from twitter data". In: *Proceedings of the 23rd international conference on world wide web.* 2014, pp. 439–444.

[103] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems* 28 (2015).