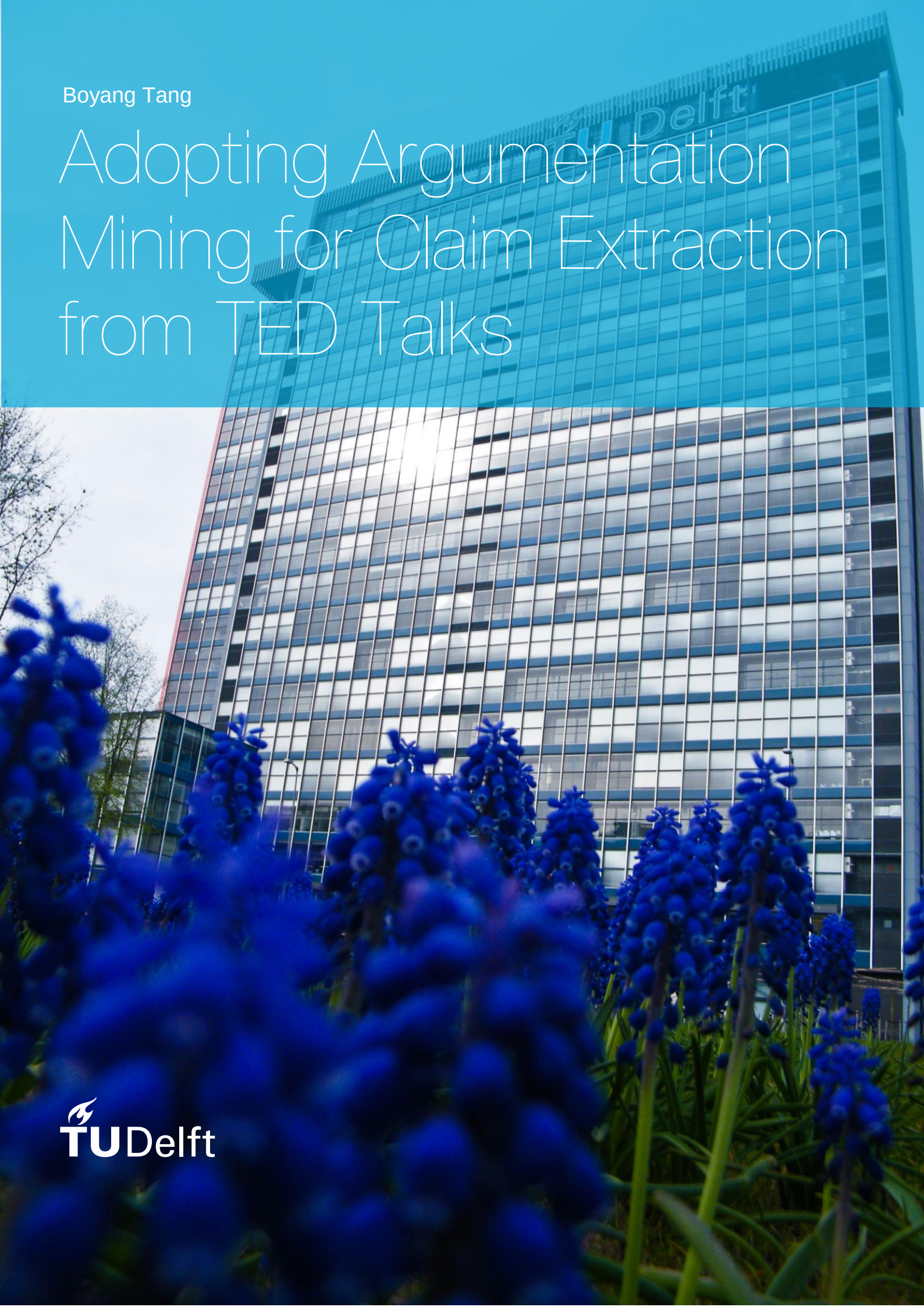
Boyang Tang

Adopting Argumentation Mining for Claim Extraction from TED Talks



TU Delft

Adopting Argumentation Mining for Claim Extraction from TED Talks

By

Boyang Tang
4518926

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science / Data Science & Technology

at the Delft University of Technology,
to be defended publicly on Wednesday October 18, 2017 at 13:30 PM.

Supervisor:	Dr. Christoph Lofi	WIS, EEMCS, TU Delft
Thesis committee:	prof. Dr. ir. Geert-Jan Houben	WIS, EEMCS, TU Delft
	Dr. Christoph Lofi	WIS, EEMCS, TU Delft
	Dr. Sebastian Erdweg	PL, EEMCS, TU Delft

TU Delft

Preface

This thesis is motivated by an AI chatbot project in FeedbackFruits. We are aiming at building a chatbot to assist students to study better. One function of this chatbot is that it can recommend TED Talks based on what students are currently learning. Since we are building a chatbot, we want to make it able to communicate with users. When recommending students, we want this chatbot also convince users to spend some time watching the recommended TED Talks. This thesis is going to check if we can achieve this goal with the cutting-edge argumentation mining techniques.

This thesis has been written to fulfill the graduation requirements of the Master Program of Computer Science at the Delft University of Technology. I was engaged in researching and writing this thesis from January to October 2017.

I cannot complete my thesis without the help of many people. First, I want to thank my supervisor Dr. Christoph Lofi. he provided me with inspiring and valuable advice and helped me a lot with finishing this project. I also appreciate the help from professor Geert-Jan Houben and Dr. Sebastian Erdweg. They gave me a lot of helpful feedback on this thesis. And I also grateful for my friends and colleagues. They also helped me a lot in finishing the crowdsourcing task and the user experiment.

Boyang Tang
Delft, October 2017

Contents

Adopting Argumentation Mining for Claim Extraction from TED Talks

Boyang Tang
4518926
B.tang@student.tudelft.nl

Abstract

Engagement is critical for academic learning. It's commonly believed that motivating students to learn is crucial in education. We think that by providing students some interesting content based on what they are learning is a good idea. Since TED Talks share attractive new ideas, we are planning to motivate students by recommending TED Talks relevant to their learning content. Also, we found it's important to have some "teasing texts", which are used to convince students to watch TED Talks we recommended. to get these texts, we are going to adopt an argumentation mining technique called "Claim Extraction" on TED Talk subtitles.Claim extraction uses classifiers trained on a dataset to extract claim sentences from the given texts. And these claim sentences can be used as the "teasing texts". Due to the fact that there isn't any TED Talk based corpus and building one is extremely expensive, we have to train classifiers on the existing Wikipedia dataset. It means we have to deal with the cross-domain learning problem. This thesis will introduce our approach of building a TED Talk claim extraction system. This system will use classifiers trained on existing corpus and can extract claim sentences from TED Talk subtitles. Also, this thesis proposes using claims extracted from TED Talk subtitles can promote students to watch the recommended TED Talks.

1 Introduction

Motivating student has always been an interesting study question to researchers and educators. While some researchers are focusing on improving structures of lectures or setting clear and attractive goals for students, previous research has also proved that we can motivate students by boosting their interest to learn. We believe that providing TED Talks to students based on the topics they are learning could boost their interest in those topics, and make them spend more effort on their studies. TED Talks are short speeches devoted to spreading ideas. We believe these interesting, powerful ideas, relevant to the learning contents, are able to boost students' interest. Thus, we are going to motivate students by recommending TED Talks related to their learning contents.

When recommending TED Talks to students, it's important and necessary to have some short "teasing texts" which can convince students that those TED Talks we recommended are indeed sharing some interesting ideas. These "teasing texts" can motivate students to watch the recommended TED Talks. A problem is: how can we find such "teasing texts"?

Generally, a TED Talk consists of one main opinion on a certain topic, and a set of evidence supporting this opinion. It is similar to an argumentation. Argumentation is the action or process of reasoning systematically in support of an idea or opinion. The idea is usually described by making a claim, which is a statement or assertion that something is the case. The claim made in a TED Talk holds the most important ideas that speakers want to share. They are the most attractive part of a whole TED Talk. Thus, we believe that using claims extracted from TED Talk subtitles as the "teasing texts" is reasonable.

Detecting claims from TED Talk subtitles manually will be too expensive since it requires too much human effort. Researchers nowadays handle this as a machine learning classification problem. They build classifiers to detect claims from a given text automatically. Classifiers will

learn features from labeled samples, and try to categorize given unlabeled samples. The training samples usually belong to a specific domain, which means classifiers are generally domain specific. Classifiers are less likely to work on data which does not belong to the same domain. However, modern approaches are mostly domain specific and focus on articles or essays that are using written language. Meanwhile, a subtitle is the direct record of the speakers' talk, which means the subtitle is closer to oral language. Oral language differs a lot from written language in many aspects. They both have their unique vocabularies and grammatical constructions. No research has done to extract claims from TED Talk subtitles which use oral language. Also, annotating enough TED Talk subtitles to build the training set will take a significant amount of effort. Building a dataset based on a large number of properly annotated TED Talk subtitles is too expensive. It cannot be done in a short period. Thus, we use an alternative solution in this thesis, that is to build a cross-domain classifier. This classifier will be trained with an existing dataset. We want to evaluate in this thesis if the classifiers we built can work properly on TED Talk subtitles.

To extract claims from TED Talk subtitles, this thesis will answer the two following research questions:

- Can we build a cross-domain claim detection system using existing datasets and approaches?
- Can the claims found in TED Talk subtitles be used as "teasing texts", and motivate users to watch the recommended TED Talks?

The most widely-used dataset in argumentation mining is a dataset built with over 1000 Wikipedia articles, published by IBM in 2015. These articles are selected under 52 different topics. It is the biggest dataset available and covered many different topics. This dataset has been used in many research. [16] [32] [1] [19] [20] [21] Thus, we consider this dataset as our first choice. Classifiers built in this thesis will all be trained on this dataset.

Usually, only a few sentences in an article or TED Talk subtitle are claims. A Wikipedia article may contain over 400 sentences but only around 10 sentences are claims. The dataset built with Wikipedia articles will be extremely imbalance. Thus, we propose that we can further improve the performance of classifiers by applying data balancing strategies.

Also, since we are looking for claims relevant to a given topic from TED Talk subtitles, we built 2 additional component according to the characteristics of TED Talk subtitles. First, we found that most of the sentences in TED Talk subtitles are long and complex. Usually, a claim is not a full sentence but only part of it. To exclude the non-claim part of a sentence, we implemented an additional component to find the sub-sentences of a given sentence. These sub-sentences are more likely to contain only the claim part of the sentence. And since we are looking for claims that are relevant to the given topic, we use a topic relatedness filter component to help filter out claims that are not relevant to the topic.

The main contributions of this thesis are:

- As a baseline, implemented several classifiers based on three existing approaches from previous research. The classifiers are trained and evaluated on the Wikipedia dataset to show that the implementation is indeed functional.
- Applied 2 different data balancing strategies to the Wikipedia dataset. Shown that applying a data balancing strategy can improve the performance of the classifiers on the Wikipedia dataset and other claim dataset.
- Evaluated the classifiers on two different datasets, shown that some of the classifiers built in this thesis can overcome the cross-domain learning problem while others can not.
- Built a TED Talk claim detection system with a cross-domain claim classifier and two additional components. Evaluated that our system can detect claims from given TED Talk subtitles. Also, evaluated if the two additional components (sub-sentence generating component and Topic relatedness filter component) can help the classifiers detect claims from TED Talk subtitles.

- Did a user experiment to check the feasibility of using claims as "teasing texts". Shown that claims extracted from TED Talk subtitles can be used as "teasing texts", and they performs better compared with texts generated by other techniques.

The following sub-sections will introduce the motivation of this thesis as well as the challenges we are facing in detail.

1.1 Students Need Motivation

Engagement is critical for academic learning. It is widely held that motivation and cognition are crucial determinations of student engagement in school. [37] Yang et al. [41] show in their research that the dropout problem is severe in the MOOC course. They took the course Duke Universities Fall 2012 offering of Bio-electricity as an example. Figure 1 shows a detailed statistic result of the student participants of this course. It turns out that although 12175 students registered at the beginning of this course, only 7761 students watched at least one video of this course and only 3658 students who took at least one quiz during the course. Also, 1257 students answered all questions in the first week, but only half of them still answered all questions until week 4. And finally, only 313 students passed the final certification. The decreasing of each stage provides substantial evidence to the idea that students do need motivation.

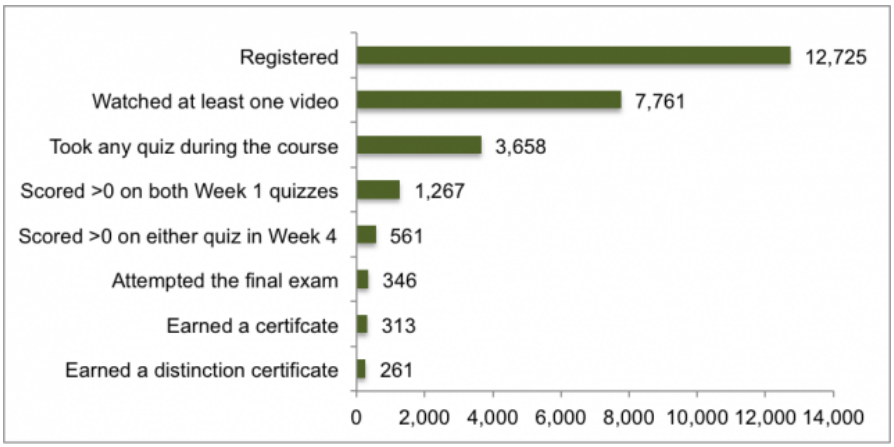


Figure 1: Student Persistence in Bioelectricity, Fall 2012(Duke University MOOC)

Motivating student has always been an interesting research question. In fact, motivation is probably the most important factor that educators can target to improve learning. [40] Researchers point out that interests and goals have been identified as two important motivational variables that impact individuals' academic performances. [14]

Having a clear goal is one of the most popular factors of motivating students and has been widely studied since the early 1990s. Many types of research have been done on different types of goal constructs and their role in motivating students. [2] Ford [9] introduced the goal content approached in one of his book published in 1992. This method assumes that there are multiple goals the students want to achieve in class. And he also provides 24 different kinds of general goals individuals might pursue in any context. By setting up short-term and long-term goals. PR Pintrich [31] did some experiments to evaluate the impact of goal orientation in self-regulated learning. It turns out that students who are good at setting goals and plans, and can monitor and control their behaviors in line with these aims, are more likely to do well in school. [31] Set goals for students, or help them set goals for themselves, have been proved to be an efficient way to motivate students.

Meanwhile, being interested in what they are learning is also a powerful motivation. Some researchers believe that by increasing students' interests to study, they will surely spend more time and effort on studying. In fact, many students struggle with the lack of interest in the learning content and then translates into a lack of motivation to learn. [13] A research on social

studies classroom held by JM Shaughnessy et al. indicates that students often are uninterested in social studies because they perceive it as a boring subject. [35] Ulrich Schiefele et al. did some research to test the impact of topic-specific interest on learners. [34] In one of their experiments, 53 students are assigned to either a high-topic-interest or a low-topic-interest group by a questionnaire. They are required to read some given content. People in the high-topic-interest group are given some articles about the topics that they are interested in. Meanwhile, in the low-topic-interest group, articles are chosen based on the topics that the group members dislike. After that, they gave all students a test including simple questions that require recalling the concrete details, complex questions about the grouping of facts or relations between points and deeper questions which require the subject to recombine or to compare various aspects of the text. The results support the fact that interest motivates the reader to go beyond the text's surface and to try to understand its meaning and main ideas.

In this thesis, we are going to use the second solution, that is, using interesting learning materials to inspire and motivate students to study. However, choosing the right learning materials itself is a big challenge. There are countless learning materials online and can be easily accessed by students. However, whether these contents can be used to motivate student are still unclear. Further research is needed to prove it. In this thesis, we suppose that TED Talks have the potential to be great materials in motivating students. The reason of choosing TED Talks will be discussed in the next sub-section.

1.2 TED Talks

TED is a nonpartisan nonprofit devoted to spreading ideas, usually in the form of short, powerful talks. [11] It is aiming at inspiring people by compelling and interesting thoughts or new ideas. For example, in a TED Talk called "Big data is better data", the speaker says that big data is going to steal our job, completely change the way we live. These ideas can draw peoples attention, inspire their curiosity and make them wants to learn more about the topic these ideas are talking about. We believe that using TED Talks as a kind of interesting contents that can improve the engagement of students. It can encourage students to spend more time to study, make them go deeper in the given materials such as research thesis or slides, and understand their meanings and main ideas.

To motivate students, the TED Talks we provided should be highly related to their current study. Recommending a TED Talks which is mainly talking about 'Building an artificial intelligence' could be hard to motivate students to learn music. By suggesting TED talks based on the study subjects they are currently learning, we can make students study more efficiently and more effectively. A recommendation system could do the job.

1.3 Detecting Claims as "Teasing Texts"

When recommending TED Talks to users, the recommendation system should also convince users to accept recommended TED Talks. It is a good idea to show some short and interesting texts which could summarize the main idea of recommended TED Talks. These texts should be able to draw users' attention, convincing them that it is worth spending time watching recommended TED Talks. We call these text "teasing texts". Although TED Talks all come with short descriptions, the description of a TED Talk will introduce the content of this TED Talk as well as its author. Some information given by descriptions is not necessary. For example. The description always has information like "Astrobiologist Armando Azua-Bustos grew up in this vast, arid landscape and now studies the rare life forms that have adapted to survive there, some in areas with no reported rainfall for the past 400 years.". It's introducing the speaker rather than the TED Talk itself. It is hard to motivate users to watch the TED Talk with this piece of text especially when users don't know the speaker. In other words, we think that "teasing texts" should focus more on the content of TED Talk.

Since TED Talks all have high quality manually generated subtitles, we can try to extract important ideas made in a TED Talk by analyzing its subtitle. Text summarization seems to be the best choice of capturing the ideas. Text summarization is the process of automatically

creating a compressed version of a given text that provides useful information for the user. [7] In other words, it summarize the main information presented in the given texts. It suits our purpose well.

Also, unlike other articles or thesis which are mainly focusing on facts, TED Talks are primarily focusing on ideas and thoughts. The basic structure of a TED Talk is similar to argumentation. Speakers will always start with one or multiple claims. And claims are supported by other more detailed claims (premises) or by evidence. In other words, TED Talks are constructed under argumentation structure. Figure 2 shows a simple and basic structure of argumentation. A claim is short. It's usually only one single sentence in a TED Talk. The evidence, on the other hand, could be several paragraphs. In the TED Talk mentioned before, the evidence is mostly talking about the speaker's experience of being a researcher in Nokia. These contents can support the claim "human insight is needed in the big data", which is the main idea of this TED Talk, but it doesn't directly relate to the topics such as "big data" or "human insight". Meanwhile, the main idea of this TED Talk can be represented well by the claim sentences. Thus, extracting claims from a TED Talk could capture the most valuable information given by this TED Talk. We believe that we can also use the claims found in the given TED Talk subtitle as the "teasing text".

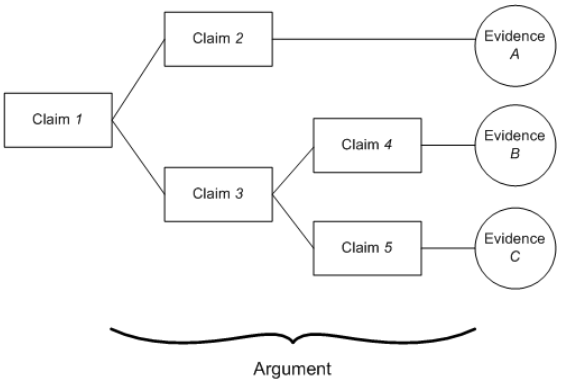


Figure 2: A simple structure of argument

In conclusion, we believe that by extracting claims from the given text, we could provide better "teasing text". Claim detection is part of the argumentation mining task, which is aiming at extracting argumentation component form the given text with computer science approaches. In other words, it requires computers to extract claims from a given text automatically. The detail of argumentation and argumentation mining will be described in section 3.

In argumentation mining, claim extraction is usually handled by a classifier that can predict if a sentence is or contains a claim. Treating the claim detection as a classification problem is the best-known solution in recent years. Researchers now are trying to get claims automatically by training a binary classifier with a large, properly-annotated corpus. Several approaches have been proved that can successfully predict if a sentence is or contains a claim. However, there are still some challenges when we are trying to extract claims from TED Talk subtitles.

1.4 Challenges in Claim Detection

We are facing several challenges when extracting claims from TED Talk subtitles. The first challenge comes to us is building a proper dataset. Classification problem always requires large amount of data as the training and testing set. And the classification algorithms are often domain specific, which means the classifiers can only work on data that is in the same domain with the training set. Thus, the best way to extract claims from TED Talk subtitles will be annotating a significant amount of TED Talk subtitles and build a classifier on this corpus. However, the annotating process is too complicated and require expert knowledge since the distinction between a claim sentence and other related texts can be quite subtle. [1] Also, in

practical, especially in a speech that contains lots of "speech-only" words such as "Well" and "OK", a claim is not always a full sentence. Sometimes, only part of the sentence is considered as a claim. For example. In the dataset published by IBM [1], the whole sentence is "However the Catalyst Model specifically states that media influences are too weak and distant to have much influence". Only the fragment "media influences are too weak" is labeled as a claim. Besides, sometimes the claim part is incomplete. For example, in the sentence "Differential treatment of racial groups that are intended to ameliorate past discrimination, rather than to harm, goes by other names". The part that is labeled as a claim is "is intended to ameliorate past discrimination", which lacks the subject. This makes the annotating process even harder. Labeling a lot of TED Talk subtitles will consume a huge among of time that makes it nearly impossible for us to finish in a short period. Also, it's hard to maintain the quality of the labeling process due to the lack of expert knowledge. Thus, this research is looking for an alternative solution. That is training a classifier with existing corpora and make it also work on TED Talk subtitles.

Another challenge is finding a suitable dataset to train the classifiers. Since argumentation mining is a relatively new research field, the number of published papers in this field are limited as well as the open-sourced dataset. The most well-known dataset that is widely used in the research field, is the IBM Wikipedia dataset used in IBM's Debater project. In this dataset, 315 Wikipedia articles chosen under 32 different topics, the topics they selected cover variety from atheism to the US responsibility in the Mexican drug wars. [1] One year later they enriched the dataset to 1289 Wikipedia articles chosen under 58 different topics. Over 80000 sentences contained in this dataset which makes it the biggest open source dataset available. However, only 2294 claims are found in all 1289 articles. This means the Wikipedia dataset is incredibly imbalanced. When training a classifier with it, the non-claim sentences can easily become dominant. Other datasets, such as the Persuasive Essays dataset published by Christian Stab and Iryna Gurevych from UKP [36], are smaller compared with IBM's dataset. Most of them contain only thousands of sentences. And some datasets like ECHR corpus focused on only one specific topic. Training a classifier on these datasets may end up with a classifier that works only on specific topics. The detail of open source data available online will be discussed in section 4.

Implementing classifiers based on approaches from previous research is also a big challenge since those approaches are described in high level. No detail about the implementation are mentioned in papers we found. Further more, the approach published by IBM relies on a language parser that is only accessible by IBM's researchers. We need to be careful when using alternative parsers. Also, we need to evaluate our implementation carefully and check if our implementations are successful.

Besides, when using the Wikipedia dataset to train a classifier but apply it on TED Talk subtitles, we may face the cross-domain learning problem since the style of writing is quite different between Wikipedia articles and TED Talks subtitles. Wikipedia articles are more like formal writing articles which mainly serve as introduction or explanation materials. Sentences in these articles are well pruned, nearly no useless information is given. Meanwhile, TED Talk subtitles are direct records of someone's speech. During a speech, the speakers may use some "speech-only" vocabularies such as "like", "Well", "OK", et al. A classifier trained on the Wikipedia articles might fail when applying to TED Talks subtitles due to these differences. Thus, the classifier should not only catch all the key features of the claim but also excludes the features that are exclusive to the training set.

Evaluating the classifiers can also be a huge challenge. Annotating a huge number of TED Talks to make our own TED Talks subtitle corpora is almost not feasible. In this research, only a small number of TED Talks will be annotated to evaluate the performance of the classifier. But testing on such a small dataset may include high bias. Meanwhile, due to the lack of expert knowledge, we must rely on crowd sourcing. The quality of annotating will be worse than the existing dataset such as the Wikipedia dataset and the persuasive essays dataset. We also need to deal with free writers who will just randomly annotate the subtitles and provides a terrible result. The crowdsourcing task should be designed carefully. A detailed introduction should be provided with the TED Talk subtitles, and quality control questions should be set to exclude free writer.

2 Related Work

Following the developing of Artificial Intelligence, some researchers are aiming at making the computer able to argue as humans do. They established a new field of study called automatic argumentation mining. The goal in this field is to automatically extract argumentation components, such as claims and evidence as well as the relationships among them, from generic textual corpora. Although studies in this field only started to appear five years ago, the growing of excitement in this area is significant. In 2014, there are three international events on argumentation mining were held including the first ACL Workshop on argumentation mining.

Research in this field starts with extracting argumentations in specific domains. In 2011, Raquel Mochales and Marie-Francine Moens [27] provided an initial approach of how machine learning and other state-of-the-art techniques can help in argumentation mining. Through their experiment on legal texts, they proved that it is possible to detect argumentation component and relations among them with general AI methods automatically. This research is the first study that is aiming at building a complete argumentation mining system although it only works on one domain of texts.

Other researchers are focusing more on building the corpora needed in argumentation mining. Hospice Houngbo and Robert E. Mercer published an approach to building a larger corpus comprising sentences that belong to specific categories of the rhetorical structure of the biomedical research text. [15] This method will not require domain expert knowledge and can represent a wider range of publications in the biomedical literature. Also, the system can distinguish the rhetorical category of sentence between four categories: Introduction, method, result, and conclusion. The conclusion sentences are usually claim sentences. However, it's hard to extend their work to build a self-annotating system on other domain since the system is highly domain specific.

Ivan Habernal et al. focused on building annotation scheme for general contents published on the web. Announced in their thesis published in 2014 that there isn't any one-size-fits-all argumentation theory to be applied to realistic data on the Web. [12] They provide two different schemes for argumentation annotation. The Claim-Premises scheme is widely-used in existing research and is the simplest way to represent the support and attack relations. Toulmin's scheme is built based on the argumentation model introduced by Stephen Toulmin [39] and is suitable for modeling static monological argumentation.

A well-known dataset for argumentation mining was built by Christian Stab and Iryna Gurevych in 2014. This study is an extended study of automated essay grading. [19] The goal of automated essay grading is to automatically assign a grade to a student's essay following several criteria. The argument structure is crucial in evaluating the quality of the essay. The dataset contains 90 essays at the beginning and has been increased to 402 essays in the second edition. These essays are annotated under Claim-Premises scheme. Due to the nature of the data, only a few sentences in each essay are non-argumentative, which makes this dataset unsuitable to be used as training set if the goal was to generalize to other genres.

In May 2014, IBM announced a new Watson project called "Debater". In the demonstration, this "Debater" can search through a large number of Wikipedia articles and come up with several claims that support or against a given topic. They also introduced that the debating technology, using in this project, is aiming at automatically extracting argumentation structures from texts in natural language. This technology could be a huge booster to debaters and decision makers to gather main points from a large number of texts in a short time and can speed up the decision-making process. This technology can be mainly divided into 3 part includes claim detection, evidence detection and mining the relationship between claims and evidence. Researchers started with the claim detection and published the first approach of the automatic claim detection system in the same year. Also, the term Context Dependent Claim (CDC) was first introduced in this research. The system built by IBM only detect claims that are directly supporting or against a given topic. Figure 3 shows the high level structure of the IBM's context dependent claim detection system. [16] This structure now becomes the fundamental structure for argumentation mining systems. The system consists three components. Sentence component takes a topic as well as its relevant articles as input, selects top 200 sentences from the given articles that are

most likely to contain claims. After that, the selected sentences are processed by boundary component. This component is aiming at extracting the best boundary for each sentence. Both sentence component and boundary component will provide a score for each sentence, indicates the reliability of the result. These scores are feed to the ranking component. The final output of this system is 50 best content dependent claims from given articles.

The sentence component in IBM's research is a classifier that can predict if a given sentence contains context dependent claims. Although some sentences may include more than one context dependent claims, it is not very common. Hence, claim detection is considered as a binary classification problem. The inputting sentence will first be encoded to extract some predefined features. It is also the most common approaches used in claim extraction systems. An English parser also built by IBM was used to obtain predefined features. Also, the system will check the sequential patterns hidden in the sentence. Researchers at IBM also believe that some sequential patterns of the sentence can be used to predict whether the given sentence containing a context dependent claim. They extended the existing sequential pattern mining algorithms to extract discontinuous patterns from encoded sentences.

This approach is extremely time-consuming. For each sentence, all the sub-sentences are checked to detect the proper boundary of the context dependent claim. A nd the sub-sentences here are all consecutive segmentations of original sentence which contain more than two words. It means a sentence with ten words will produce 36 sub-sentences. According to their research, each sentence in Wikipedia articles spans on average 23 words and around 200 sub-sentences will be generated during the process. Each of these sub-sentences will be encoded and predicted. Thus, processing a single article could take several minutes. Also, although contextual information is proved to be extremely powerful in building accurate predefined features, using this information will surely limit the generalization capabilities of the argumentation mining system. As a matter of fact, domain-specific and highly engineered features are likely to over-fit the data they have been constructed on. [20]

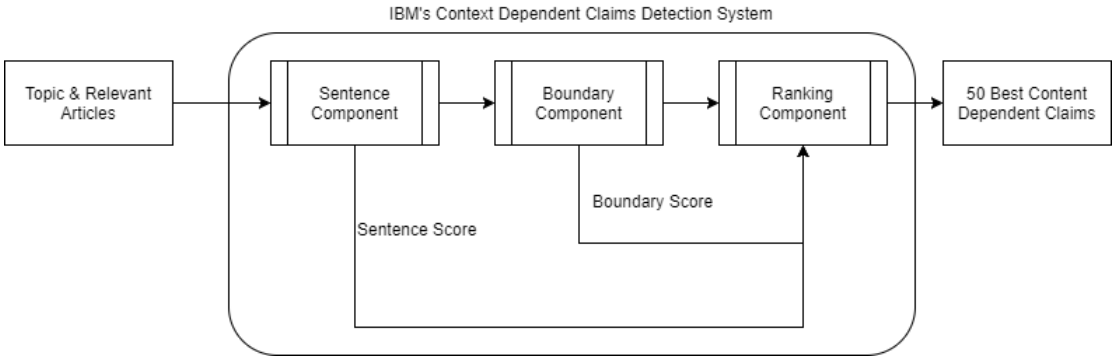


Figure 3: The high level structure of IBM's context dependent claim extraction system

To overcome these issues, researchers start constructing systems to detect sentences containing context independent claims. In 2015, Lippi et al. [19] published in their thesis a context independent claim detection system. Instead of using predefined, well-engineered, context based features extracted from the given sentences, this system takes the constituency parse trees of a given sentence as input, and Tree-kernel support vector machine algorithms are applied. Lippi et al. found that sentences containing claims are similar in some parts of their parse tree. Therefore, by examining the similarity between the parse trees, sentences containing context dependent or context independent claims can be extracted. Also, tree-kernel support vector machine could automatically construct an implicit feature space. Therefore, no predefined features are needed. And the sub-sentences can be automatically checked by using sub tree-kernel, sub set tree-kernel or partial tree-kernel. This approach could significantly speed up the prediction process.

3 Argumentation, Argumentation Mining and Claim Extraction

In order to generate some effective "teasing texts" automatically, we are going to extract claims from TED Talk subtitles. Since claim is one of the most important components in argumentation, this thesis is highly relevant to argumentation mining. More specifically, this thesis is focusing on the first step of argumentation mining, the claim detection. There are several approaches to detect claims from given texts and have been proved to be extremely powerful. Also, we use a different definition of the claim to fit our demand of generating "teasing texts". Therefore, this section will introduce the definition of argumentation, the definition of claims we used in this thesis as well as three common approaches of claim detection system that will be used as baseline systems

3.1 Definition of Argumentation

Argumentation is a branch of philosophy that studies the act or process of forming reasons and of drawing conclusions in the context of a discussion, dialogue, or conversation. [10] Being an important element of human communication, arguments are frequently used in texts as a means to convey meaning to the reader. The twentieth-century British philosopher Stephen Toulmin [39] noticed that good, realistic arguments typically will consist of six parts:

Data Data is the facts or evidence used to prove the argument. The first step to establishing an argument is to have some information that justifies it.

Claim Claim is the conclusion to be established by the argument, it is the statement that being argued.

Warrants Warrants is the supporting step between data and claim. It justifies the leap from data you provide to the claim you made.

Backing Backing, also known as backing to a warrant, which is the statements that serve to support the warrants. These statements don't necessarily support the main point or claim you made, but it does provide that the warrants are true.

Qualifiers Qualifiers are statements that define certain conditions. Qualifiers indicate that under which conditions the argument hold true.

Rebuttals Rebuttals are counter-arguments or statements indicating circumstances when the general argument does not hold true.

This structure is detailed but can be relatively too complicate for an argumentation mining system. Researchers from IBM set up a simplified structure of argument and used it in their argumentation mining research [16] [17] [1]. In their structure, an argument only contains 2 parts. They are claims and evidence. They group the data, warrants and all other components of Toulmin's model, except claim, as evidence. This structure is easier and clearer for argumentation mining approaches since it's easy to convey to human annotators. [16] In this thesis, we will also use the definition set by IBM.

3.2 Argumentation Mining

Argumentation mining is the task of identifying argumentation components, along with their relationships, from text. [10] In recent years, there has been a growing interest in argumentation mining research field. Researchers in IBM starts a project called the "Debater" project whose goal is to assist humans to debate and reason. Also, many other researchers are trying to support the decision making process with this technique. [8] When using the simplified argument structure from IBM, the argumentation mining can be mainly divided into three parts. Claim extraction is focused on detecting the claims from a given topic. Evidence extraction which will find

evidence that supports or against a claim. And Relationship mining is aiming at constructing the complete argumentation structure of a given text by finding out the support and against relationships among argumentation components.

As we discussed before, we are looking for the main ideas of the TED Talk contained in the subtitles. Therefore, we only interested in extracting the claims. The next section will only introduce the definition of claim.

3.3 Definition of Claim

In general, a claim is defined as a statement or assertion that something is the case, typically without providing evidence or proof. Cambridge English Dictionary explains that claim is to say that something is true or is a fact, although you cannot prove it and other people might not believe it. IBM research group setup a fine-grind definition of Claims. In their research, only context dependent claims (CDC) are considered as claims.

A context dependent claim is a general, concise statement that directly supports or against the given topic. [16] And a topic, defined by IBM, is a short and usually controversial statement that defines the subject of interest. [16] In this research, we will use this definition considering the use case of the claims we mentioned before. Since the "teasing texts" should be highly relevant to topics students are learning, the definition of claim from IBM suits this purpose much better. Thus, in this research, we are focusing on context dependent claims. Also, the topic, in this case, is more likely to be just a simple word or noun phrase rather than a controversial statement. In that case, we change the definition of a topic to a noun phrase or even a single word.

3.4 Approaches to Detect Claims

The most well-known solution to extract claim is the machine learning approach which treats this problem as a binary classification problem. Figure 4 shows a general procedure of the claim detection process. Both topic and candidate sentence are input into the classifier, the classifier then decides if this sentence containing context dependent claim or not. The output of the classifier is a binary value indicates if the sentence contains a context dependent claim. Some of the existing approaches are focusing on extracting context independent claim. These approaches do not require the topic as an initial input. Thus, the structure could be even more straightforward. Currently, there are three categories of most popular approaches.

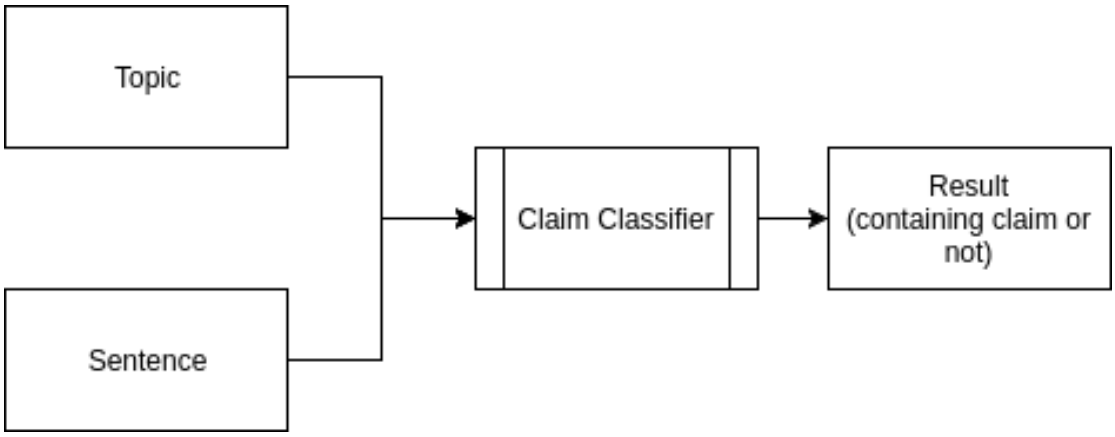


Figure 4: A simple procedure of the claim classifier.

Researchers from IBM believe that claims should have some unique content dependent characteristics. For example, a claim should always be emotional. When making a claim, it usually will support or against a certain topic, which means the sentiment of a claim is more likely to be either positive or negative, but not neutral. In addition, a claim is usually a subjective sentence because it is about someone's opinion. Also, since the claims are talking about certain

topics, the subject of a claim, which is usually the thing that a claim is talking about, should be highly relevant to the given topic. Thus, it is possible to extract some predefined features from a sentence and feed these features to the classifier. This is currently the most popular approach and was widely-used in argumentation mining studies. [8] [16] [33]

Also, IBM indicates that there are some unique patterns hidden inside the claims. When people are making a claim, there are several distinct but unique ways that people will use. The simplest example could be "something is good". There will first be a word that is correlated or directly mentioned in the topic, followed by some adjectives. Also, some words are used more frequently in claims. For example, the words "argue", "think" and "believe" are more likely to be used in claims. Thus, we can extract some unique patterns from claims. They will have the power to identify the claim sentences. [16]

Finally, some researchers believe that the structure of a sentence could be highly informative for argumentation mining, and in particular for the identification of a claim. [19] For example, Lippi et al. use the constituency parse tree to identify the claim sentence. The figure below shows a case of 2 claim sentences talking about different topics but having a similar structure. Similar parts are highlighted with the square box. These 2 sentences are quite similar to each other in most parts. Therefore, building a classifier with constituency parse trees indeed make sense. And the most suitable classifier for this task is a Support Vector Machine that uses a tree-kernel. The tree-kernel is aiming to capture the similarity between trees.

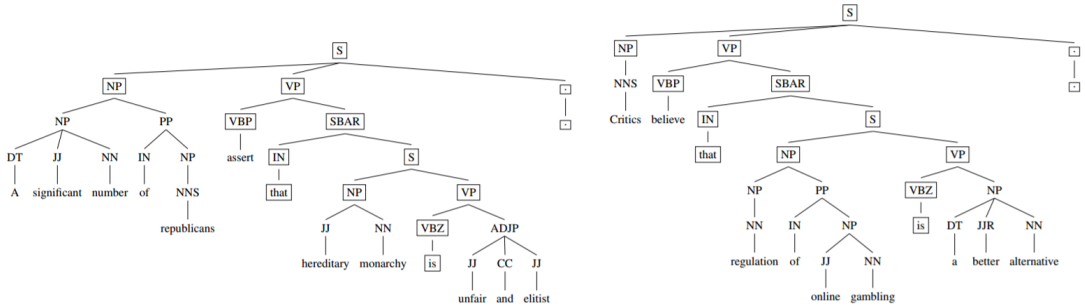


Figure 5: An example of the argumentation structure [19]

All three approaches are proved to be incredibly powerful when used to extract sentences containing claims from Wikipedia articles. They were built and tested on the Wikipedia dataset published by IBM and performed well. Therefore, in this research, we are going to implement classifiers based on these three approaches. However, these approaches are designed only to work on Wikipedia articles and none of them is tested on texts from different domain such as TED Talk subtitles. It is not a surprise that some of them will suffer the cross-domain learning problem. Thus, each classifier we built should also be validated by datasets that are created with texts from domains that are different with training domain. Thus, except the Wikipedia dataset which is used as the training set. Some other datasets are also used in this thesis. They are built on texts that are different with Wikipedia articles. The next section will introduce the datasets used in this thesis.

4 Data and Corpora

Data is crucial in any machine learning problem. As we introduced before, classifiers are often domain specific since they learn unique features from the training data which belong to a specific domain. The best way of building a TED Talk claim classifier is to train it on a dataset built with a large number of properly annotated TED Talk subtitles. However, we have discussed before that this is impossible since annotating enough TED Talk subtitles is too complicated to be done in a short time. The alternative solution is to build a cross-domain classifier, which is trained on existing large dataset with high quality. Also, in order to test it cross-domain

performance, we may need data from other domains. Therefore, several datasets are used in this thesis.

This research highly relies on the existing datasets. To explain the reason for using these datasets as training or testing sets, this section will introduce the dataset used in this thesis and compare them with other argumentation mining corpora.

4.1 IBM Wikipedia Dataset

There are several dataset available for claim detection. IBM research group published a dataset in 2014 [1] and improved it in 2015 [32]. The first version of the dataset contains 315 Wikipedia articles chosen under 32 different topics. There are more than 40000 sentences in total in this dataset and 1388 claims found in these datasets. The data contains an excel file which listed all the claims, their relevant topics and the name of the articles. Two versions of claims are provided including original sentences that contain claims, and a manually corrected version of the claims. Also, all the Wikipedia articles are provided in .txt format. However, this dataset suffers from the encoding format problem. When reading these claims with Python, around 200 of them threw the Unicode escape error.

Table 1: An example of claims in IBM 2014 dataset

Topic	Article	Claim	Correction type	Corrected Text
the sale of violent video games to minors	Video game controversies	exposure to violent video games causes at least a temporary increase in aggression and that this exposure correlates with aggression in the real world	NA	Exposure to violent video games causes at least a temporary increase in aggression and this exposure correlates with aggression in the real world
the sale of violent video games to minors	Nonviolent video game	a high degree of relationship between violent games and youth violence	VERB ADDITION	A high degree of relationship between violent games and youth violence has been indicated
the use of performance enhancing drugs in professional sports	Use of performance-enhancing drugs in sport	around 10,000 former athletes bear the physical and mental scars of years of drug abuse		

In 2015, IBM researchers published an improved version of the 2014 Wikipedia dataset. [32] In the new dataset, 1289 Wikipedia articles are selected based on 52 different topics. It contains over 80000 sentences and 2294 claims extracted from these articles. Also, all the claims are stored in a .txt file, and all encoding format problems are solved. Table 2 shows some claims given by the 2015 IBM dataset. The topics in the new dataset are changed into "This house believes" format which is commonly used in debating competitions. Also, some columns such as "Require correction" and "Correction type" are removed since they will not be used. The claims

are also provided in 2 versions. However, unlike the old dataset, the claims that don't need correction will also be given in 2 version: the corrected version and their original form. Except the two versions are the same sentence. The third row of the example table is an example of a claim that doesn't need correction. Also, the relationship between topics and articles are given in another .txt file. This file indicates the topic of each article. Thus, we can easily find the topic of an article. This dataset is mainly used as the training set in this research.

Table 2: An example of claims in IBM 2015 dataset

Topic	Corrected Text	Original Claim
This house believes that the sale of violent video games to minors should be banned	Exposure to violent video games causes at least a temporary increase in aggression and this exposure correlates with aggression in the real world	exposure to violent video games causes at least a temporary increase in aggression and that this exposure correlates with aggression in the real world
This house believes that the sale of violent video games to minors should be banned	a high degree of relationship between violent games and youth violence	A high degree of relationship between violent games and youth violence has been indicated
This house would permit the use of performance enhancing drugs in professional sports	around 10,000 former athletes bear the physical and mental scars of years of drug abuse	around 10,000 former athletes bear the physical and mental scars of years of drug abuse

4.2 Persuasive Essays Dataset

Christian Stab et al. [36] published an improved version of their persuasive essays dataset this year. This dataset collects 402 different persuasive essays and contains 4000 sentences in total.

Persuasive essays are essays that show someone's opinions about certain things. For example, one essay in this dataset shows the author's opinions about whether competition benefits students more than cooperation. These essays also focus on ideas and thoughts instead of the fact, which makes them more similar to a TED Talk and different a lot from Wikipedia articles.

Also, the claim sentences found in this dataset are different from claims in the IBM's Wikipedia dataset. This dataset uses the standard definition of the claim since it is focusing on finding the relationship between claims. Although the persuasive essays do have specific topics associated with them. The sentences labeled as claims are not required to be relevant to a specific topic. Sentences that meet the standard definition of the claim are marked as claims.

In this dataset, some sentences are labeled as "premise". According to the Cambridge English Dictionary, a premise is an idea or theory on which a statement or action is based. In other words, a premise can be treated as a claim that supports another claim. In TED Talks, speakers usually made multiple premises to support their main claim, which means the premises also hold some crucial information about the main idea of the TED Talks. Thus, in this research, the premises are also considered as claims.

The persuasive essays dataset is only used as the test set in this project to check the performance of the classifier on a different type of texts. In other words, it checks the cross-domain performance of the classifier. The reason of using the persuasive essays dataset is that They are similar to TED talks since both of them are focused on showing someone's ideas and thoughts. And, like TED Talk subtitles, the sentences in those essays are often complex sentences. Also, over 4000 sentences in this dataset make it the second largest dataset available.

In the persuasive essays dataset, the essays don't have topics associated with them. Yet, each of them still has a proper title which can be used as the topic of this essay. In this thesis, we will use the title as the topic of an essay. However, this dataset has one fatal flaw. The

claims in this dataset are context independent claims. Which means the claims extracted from an essay are not necessarily supporting or against the main topic of this essay. For example, in an essay named "Roommates quality and their importance", the sentence "Communication is very important" is labeled as "Claim". Although it is not relevant to the topic. Therefore, when using this dataset as the testing set, the recall is not so reliable.

4.3 TED Talk Subtitle Dataset

Evaluating the performance of the classifier on extracting sentences containing context dependent claims from TED Talk subtitles is crucial in this research. It is necessary to evaluate them on a dataset created with properly annotated TED Talk subtitles. Due to the fact that there isn't any dataset that is built with TED Talk subtitles, we are going to build a small TED Talk dataset for the purpose of testing. According to the previous research on annotating the corpora, building a relatively large dataset may take years. Therefore, in this research, only 10 TED Talk subtitles are used. The dataset will be relatively small compared with existing datasets. However, we believe that by selecting the TED Talk subtitle under different topics and combine the result tested on this dataset with results on persuasive essays datasets, we can draw a solid conclusion on the cross-domain performance of the classifiers we built.

4.4 Other Dataset

Apart from the three datasets mentioned before, there are more published datasets that could be used in this project. However, these datasets all have some issues and cannot be used in this thesis. For example, Mochales Palau and Moens built a corpus with legal texts [28]. This corpus only contains claims relevant to a specific domain. Classifiers built with this corpus are hard to be extended to work on TED Talk subtitles since the dataset is quite small and the legal texts are too unique which differ a lot from TED Talk subtitles. Cabrio and Villata [3] published a corpus created on Debatepedia pages. Although the articles are chosen under various of topics, this corpus doesn't provide any false-label data. That is, all sentences in this corpus are claim sentences. Peifeng Li et al. [18] implemented a argument extraction model on ACE 2005 Chinese corpus. The corpora they used built with Chinese news articles. It cannot be used to train or test classifiers in this project since we are targeting on English texts.

5 Claim Detection

Claim detection is done by training a classifier that can predict if a given sentence contains context dependent claims. Three different approaches for detecting sentence containing context dependent claims are introduced in previous research. They have already been proved that perform extremely well on Wikipedia articles. We think it is a good idea to build classifiers based on these three approaches.

In this thesis, three different kinds of approaches are implemented. Sentence component feature classifiers are classifiers that deals with predefined features extracted from sentences. Sequential pattern classifiers can classify sentences based on some sequential patterns found in claim sentences. And tree-kernel support vector machine classifiers can find claims by checking the constituency parse tree of the input sentences. As we introduced in section 3, these three kinds of approaches are well-known and widely-used in argumentation mining system. And they are proved to be incredibly powerful in extracting claims from Wikipedia articles. Researchers have spent several years to validate and improve the performance of these approaches. However, researchers only validated the performance of these approaches using the IBM's Wikipedia dataset. No research has been done to prove that these approaches are able to overcome the cross-domain learning problem. Thus, it is a good idea to implement classifiers based on these methods, try some different techniques such as different classification algorithm and check if they can overcome the cross-domain training problem.

5.1 Sentence Component Feature Classifier

5.1.1 Procedure

The first category of classifiers built in this thesis is based on the IBM's ideas about extracting sentence component features. [16] That is, some predefined, content based sentence component features can be used to detect the claims. The research papers published by IBM roughly provide a general idea about what kinds of features should be utilized. Some of the features are extracted with a private parser also built by IBM research groups that cannot be used by other researchers. This thesis provides a detailed alternative way of the implementation of the feature extraction process with an open-source, well-known natural language processing tool kit, that is, the CoreNLP took kit published by Standford University. [22]

Figure 6 shows a detailed procedure of this classifier. When given a sentence and topic, the feature extraction component will extract 11 pre-defined features out from the given sentence and the given topic. After that, each sentence is presented in an 11 dimensions vector. This vector is passed to the classifier to get the predicted label. The classifier will return a score between the interval $[0, 1]$. This score is the possibility that the input sentence contains context dependent claim. We can translate it into a binary value by setting up a threshold. The binary value indicates whether the sentence containing context dependent claim. In this research, the threshold we use is 0.5.

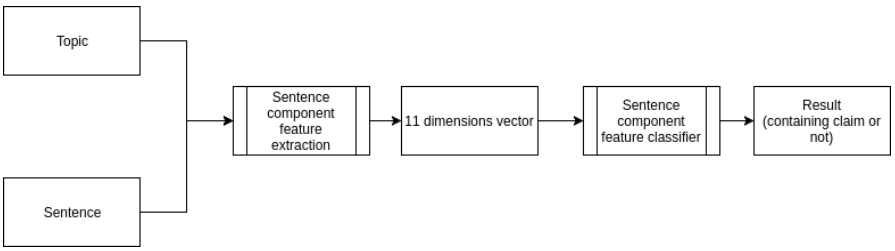


Figure 6: A detailed procedure of the pre-defined feature classifier.

The 11 predefined features can be divided into four different categories. Topic relevant features will catch the relevant between this sentence and given topic. Vocabulary and grammar features can find out the unique vocabularies or grammars used in the claims. Sentimental and subjectivity features is aiming at indicating the sentiment and subjective ratios of the sentence. And finally, the sentence length feature is going to count how many words does this sentence have.

5.1.2 Topic Relevant Features

Topic relevant features are used to indicate the relatedness between topic and sentence. Since I'm looking for the context based claim which should be highly relevant to the topic, these features indeed have the power of detecting the claims we want. Topic relevant features include:

Subject relatedness Calculate the relatedness between the topic and the subject of the candidate sentence. if multiple subjects detected in the sentence, use the one with the highest relatedness.

Synonyms relatedness Calculate the relatedness between the synonyms of the subject and the topic. If there are multiple synonyms, one with the highest relatedness is used

Hypernyms relatedness Calculate the relatedness between the hypernyms of the subject and the topic. If there are multiple hypernyms, one with the highest relatedness is used

Hyponyms Calculate the relatedness between the hyponyms of the subject and the topic. If there are multiple hyponyms, one with the highest relatedness is used

Noun words or phrases relatedness Calculate the maximum relatedness between sentence and Noun words or phrases in the sentence. The subject is not included in this feature. This feature solves the problem when the subject is either missing or a pronoun.

In this category of features, the relatedness between sentences and topic are represented by the subjects and their expansions between topic. It is because the relatedness is calculated with word2vec technique. When training the word2vec model, words that are frequently used together will be considered as highly relevant. For example, the relatedness score between "YouTube" and "computer games" is 0.42, which means the term "YouTube" is not so relevant with "computer games". But the relatedness score between the sentence "YouTube is the largest video website" and "computer games" rise to 0.56 because the relatedness score between "video" and "computer" is high (0.67). Although it's clear that this sentence is not talking about computer games at all, it will be considered as "weakly relevant" to "computer games" by the system since the relatedness score is over 0.5. Thus, including the overall relatedness between sentence and topic will lose the information about why the sentence is relevant to topic, which is considered as one of the most important information to extract claim by IBM's researchers [16]

In IBM's research, these features are extracted using ESG parser [24]. And there is a group of features called "ESG features", which are some binary features extracted by the ESG parser [16]. However, this parser is not published which means there is no way to use this parser in this research. Instead, these features are extracted using Standford CoreNLP tool kit. And the most important parser used in this research is the Enhanced++ Dependencies parser.

Enhanced++ Dependencies parser is used to extract the subjects from a sentence. Figure 7 shows an example of the output of Enhanced++ Dependencies parser. The subject word is marked as "nsubj". That is the word "game" in this example. However, some adjective words should also be part of the subject. In this case, the ideally subject is "violent video games." Thus, we also extract the words that have "compound" or "amod" relationship with the subject word. In this case, there are four subjects obtained in total. they are "games", "video games", "violent games" and "violent video games".

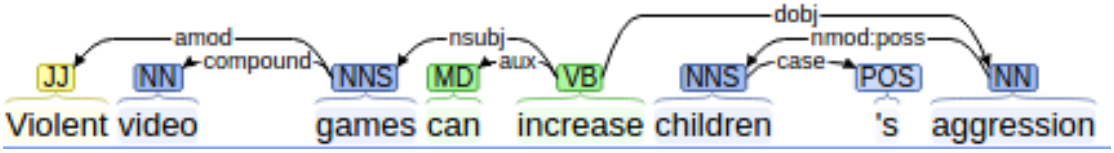


Figure 7: An example of the output of Enhanced++ dependency parser.

Also, a subject is expanded with its synonyms, hypernyms, and hyponyms. Hypernyms are words above the subject. In other words, hypernyms are broader than the subject. Hyponyms are words that below the subject, which means they are more detailed compared to the subject. And synonyms are words that are similar to the subject. For example, in this case, we have subject "video game", the synonym can be "computer game" which shares the same meaning with "video game". The hypernym is "game" since it contains "video game" and another type of games such as poker. And the hyponyms can be "virtual reality". Figure 8 provides an example of hypernyms, hyponyms, and synonyms.

In this research, WordNet is used to expand the subject extracted from a sentence. WordNet is an English lexical database first published by George A. Miller in 1990. [26] This database group English words into sets of synonyms called synsets. It also provides the relations between these synsets, including hypernyms and hyponyms that are used in this research.

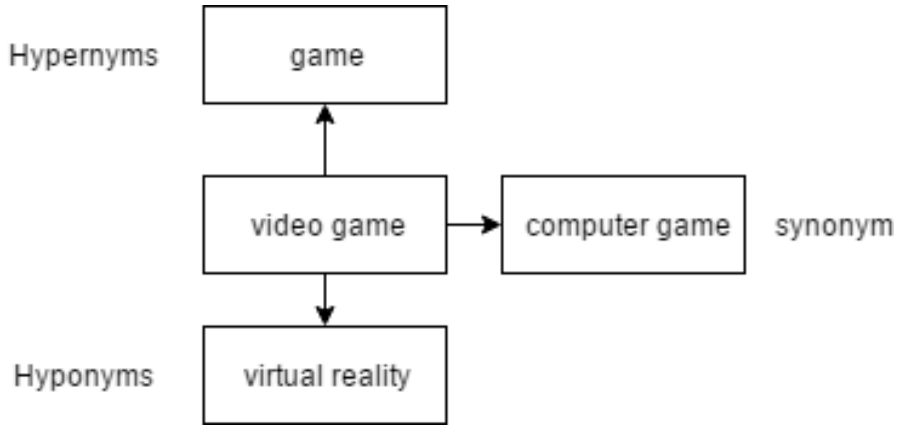


Figure 8: An example of synonyms, hypernyms and hyponyms.(the result is given by the NLTK python natural language processing package)

Word2vec is used to calculate the relatedness between 2 phrases. Word2vec is a group of related models that are used to produce word embeddings. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. In other words, a word will be transformed into a high dimensional vector by word2vec model. [25] By calculating the cosine similarity between two vectors, we can get the relatedness between these two words.

However, in this research, topic and subject can also be noun phrases. We need to calculate the relatedness between multi-word terms. A pre-trained word2vec model cannot be expanded to support phrases easily without redo the whole training process. Also, the size of the vector space will increase a lot even if we only consider the bigram phrases. To solve this problem, IBM's researchers provided an alternative approach that can calculate multi-word term relatedness. [17]

When comparing the relatedness between 2 phrases, namely $P_1 = \{W_1, W_2, ..., W_n\}$ and $P_2 = \{W'_1, W'_2, ..., W'_n\}$. First take a word W_x from P_1, calculate the relatedness between it and all words in P_2. use the highest relatedness r_{xy}. Then, iterate all word in P_1 and use the average relatedness of all highest relatedness r_{xy} as the relatedness between 2 phrases. Figure 9 shows and example of calculating the relatedness between "violent video games" and "computer games"

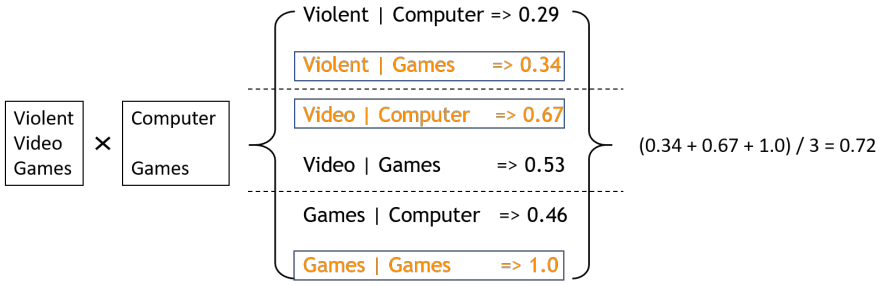


Figure 9: Calculating the relatedness between "violent video games" and "computer games"

Sometimes the claims may be inverted sentences, for example, "The cause of children's aggression is violent video games". In this case, the subject is "the cause" instead of "violent video games". However, this term that we are looking for does appear in the sentence. Thus, the relatedness between other noun word or phrases in the sentence and the given topic are also checked. The subject is not included since it has already been checked.

5.1.3 Vocabulary and Grammar Features

The vocabulary and grammar features are aiming at catching the unique usages of words or grammars in the claims. When people making a claim, there could be some specific grammars that are commonly used. For example. People may start their claims with "I believe that". Thus, the appearance of the word "that" that is used as the conjugate word might be a useful feature when detecting claims. Also, a claim is less likely to mention specific year or location. In fact, in IBM's research [16], these features have been proved to be crucial features and can distinguish the claims by their grammar and vocabulary aspect.

The features in this group contain:

Conjugate that A binary feature indicates if there is a word "that" in this sentence which functions as conjugate words.

Verb in present A binary feature indicates if there is a verb in this sentence that is in its present format.

Years A binary feature indicates if there is a specific year mentioned in this sentence.

Location A binary feature indicates if there is a specific location mentioned in this sentence.

5.1.4 Sentimental and Subjectivity Features

The sentiment of a claim usually will not be neutral. When people are claiming something most of the time, they will mix their own opinions into the claim. A simple example could be "Video game is bad for children." which is mostly negative. In addition, unlike the other content such as the introduction or short stories which are mainly focused on the facts, claims should be more subjective since it's about showing someone's opinion. These two features should be the unique features of claims.

The feature in this group contains:

Subjective score A score between 0 to 1 indicate how subjective this sentence is. 1 indicates that the sentence is completely subjective and 0 means completely objective.

Sentiment ratio A score between 0 to 1 indicate how neutral this sentence is. 1 means the sentence is completely neutral and 0 mean it's completely sentimental.

The subjective score is given by a classifier trained on NLTK corpus. NLTK provides a corpus for training the subjectivity score classifier. This corpus is built on a large number of tweets. Each tweet is labeled as either "objective" or "subjective". NLTK also provide a built-in model that can extract features from these tweets and train a classifier that can predict the subjectivity score of a given sentence.

In addition, NLTK also provides a model that can perform sentiment analysis. The output contains 3 scores: positive score, negative score and neutral score. the sum of 3 scores always equal to 1. We only use the neutral score since we don't care the sentence is positive or negative.

5.1.5 Sentence Length Feature

Sentence length feature is simply the number of words in the sentence. Claims usually are short sentences compare with other content. People may use long sentences when they are telling a story or are explaining some complex terms, but a claim is usually a short, clear sentence that shows someone's opinion only. By checking the claims in IBM dataset, we found that most of the claims only contains around 15 words and is much shorter compared to other sentences. Thus, this feature can be used as a rough indicator of the claim. Sentences that are too long are less likely to be claim sentences.

5.1.6 Classification Algorithms

Several kinds of classification algorithms can be used. The Most commonly used algorithm is Support Vector Machine. Nearly all researchers tried to build an SVM classifier to detect claims. In IBM's research, [16] Researchers used the Logistic Regression classification algorithm due to its efficiency and its model interpretability. [16] However, there are still a lot of options available. And the performances of other classifiers are unclear. No research has compared the performance of different classifiers on this task yet. And IBM didn't prove in their research that Logistic Regression is the best choice. Thus, it's a good idea also to try other algorithms

When checking all 11 features extracted from the sentence, we found some of them are not that absolute. For example, most of the time a claim may not include the year. However, "There will be infinitive food in the year 2020" is also a claim that includes a specific year. Also, some features may be used before others to filter out most of the unwanted sentences. For example, we can first check the subjectivity and sentimental of the sentence. If a sentence is neutral and objective, the probability that it contains a context dependent claim will become extremely low. Thus, the decision tree may fit this job better. In this thesis, we decided to use the extreme gradient boost (Xgboost or XGB). [5] since it's one of the cutting-edge tree boosting system.

K nearest neighbor (KNN) algorithm is also a popular classification algorithm that is used in many classification problems. However, this algorithm is never used in any research done before. This thesis will also check the performance of KNN algorithm.

The output of the classification algorithm is a fraction ranged from 0 to 1, indicates the possibility that the candidate sentence contains a claim. The sentence whose probability is larger than 0.5 is considered as a sentence that includes a claim.

5.1.7 Data Balancing

The IBM's Wikipedia dataset is incredibly imbalanced. Researchers managed to find 2294 claim sentences from over 80000 sentences. The number of non-claim sentences is around 35 times larger than claim sentences. Using imbalanced data directly with Xgboost and KNN will cause serious problems. Data balancing is required to get a better result since the performance of machine learning algorithms is typically evaluated using predictive accuracy. [4] Although IBM's research didn't mention that data balancing is necessary for this work, we found in this research that using balanced data can indeed improve the performance of the classifiers. The comparison of the performance between balanced and imbalanced data will be shown in later section. There are two different data balancing method used in this thesis. The main difference between these two methods is whether to create new data.

Synthetic Minority Oversampling Technique (SMOTE) is an oversampling approach in which the minority class is over-sampled by creating "synthetic" examples. [4] This technique is inspired by a successful technology used in hand writing recognition. By rotating and skewing the original picture, researchers can create more training data using existing hand writing data. SMOTE algorithm will produce synthetic samples based on the data from minority class samples. The synthetic examples cause the classifier to create larger and less specific decision regions. [4]

Also, instead of increasing the minority class samples, another kind of methods called under sample are also widely-used by researchers. Tomek Link is one of the most famous under sample algorithm. Tomek link can be considered as links between the items from the edge of 2 different classes. It can be used as a method of guided under sampling where the observations from the majority class are removed. [6].

A popular approach to dealing imbalanced data is to combine SOMTE and Tomek Link algorithm. That is, generating samples for minority class using SMOTE algorithm, and using Tomek link to ignore the items from majority class that lands at the very edge of the category. It can also make sure that the synthetic sample generated by SMOTE algorithm will not fall in the majority class. Figure 10 illustrate an example of using SMOTE+Tomek method to balance the dataset.

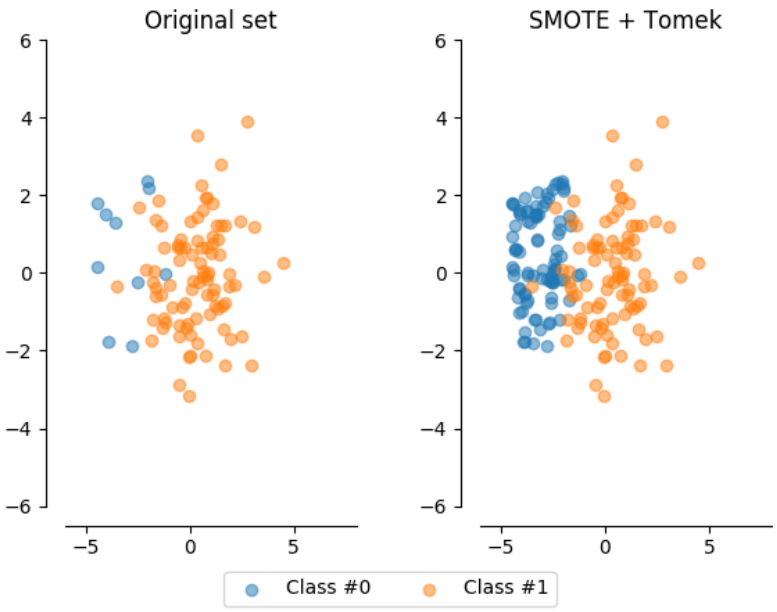


Figure 10: An illustration of the SMOTE+Tomek method.

Another approach is not to generate fake data. It can be done by using grouping strategy. By dividing the majority class into several sets, each set contains the same among of data as the minority class. We can build some subset by combining the minority class with part of the majority class. In this thesis, we randomly divide the non-claim sentences into 27 parts. Each part contains 3000 sentences. 27 different classifiers are built on the sub dataset constructed by combining the 2294 claim sentences and 3000 non-claim sentences. Figure 11 shows the structure of this strategy. Each classifier will provide a predicted result individually. The result is the combination of the result from each classifier. In this case, all classifiers are binary classifiers. Thus, the result is the majority vote of all classifiers.

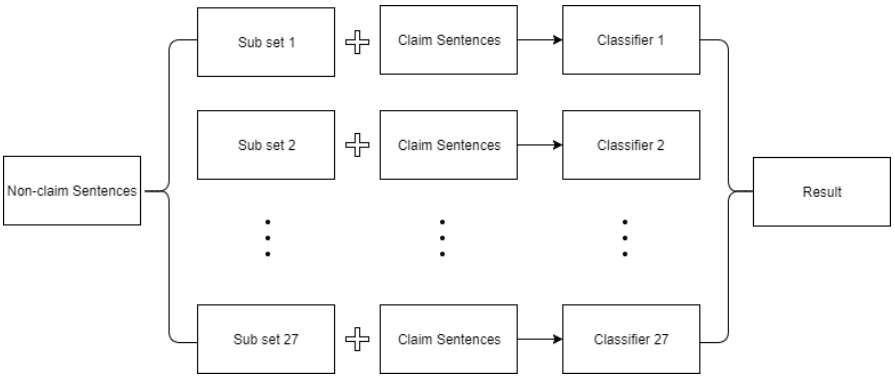


Figure 11: The structure of classifier using group strategy

Following this approach, 4 classifiers are built. They are:

SCF-XGB-SMOTETomek Sentence component feature classifier using Xgboost algorithm and SMOTE + Tomek data balancing strategy

SCF-XGB-group Sentence component feature classifier using Xgboost algorithm and grouping data balancing strategy

SCF-KNN-SMOTETomek Sentence component feature classifier using KNN algorithm and SMOTE + Tomek data balancing strategy

SCF-KNN-group Sentence component feature classifier using KNN algorithm and grouping data balancing strategy

5.2 Sequential Pattern Mining Classifier

In IBM's research [16], researchers indicate that there are some distinct patterns hidden in the claim that have the potential to indicate if the given sentence is a claim. In their research, they combine these sequential pattern mining features and pre-defined sentence component features together to predict the given sentence. However, in this thesis, we found that using the sequential pattern mining classifiers solely can already get a good performance. Using them together can't improve the performance significantly. Also, like the sentence component classifier, the research thesis published by IBM [16] only provides a general thought of the sequential pattern mining process. In fact, it only described an ideal outcome of this process. This thesis will introduce the procedure and techniques used in each step of the sequential pattern mining process.

5.2.1 Sequential Pattern Mining

Sequential pattern mining deals with data represented as sequences [23]. There are several popular algorithms that can be used to mining the sequence such as PrefixSpan and GSP algorithm. These algorithms are designed to extract the continue sequences. For example, a simple sequence data $\{d_1, d_2, d_3\}$, the sequence with a minimum length of 2, extracted by these algorithm will be $\{d_1, d_2\}$, $\{d_2, d_3\}$ and $\{d_1, d_2, d_3\}$. A sentence can be considered as a sequence of words. However, when extracting sequences from texts, since there are some words, such as adjectives, which are not that important, we will need to ignore some of the words. That is, we will need to extract pattern like $\{d_1, d_3\}$.

5.2.2 Encoding

The first step of extracting the frequent patterns would be encoding the sentence. Using the words directly will cause domain specific problem. The corpus used in the thesis can't cover all topics. When given an article whose topic is not covered by the corpus, the system will fail to detect the claim as some word in the candidate sentence may never appear in the corpus. A popular idea of encoding is using the Part of Speech Tags (POS Tags). IBM's research indicates that other features may also hold the power of indicating the claims. Thus, each word in the candidate sentence will be encoded as a tuple of indicators includes:

Word This is simply the word itself. There might be some words that are frequently used in claims. Thus, include the word itself in the indicator is necessary.

POS Tags CoreNLP tool kit can be used to extract the POS tag of each word. This indicates the words as nouns, verbs, adjectives, adverbs, et al.

Sentiment words This is a binary indicator ("Sentiment" or "None") which indicates whether the word is a sentiment word. It includes both positive and negative words

Claim words This is a binary indicator ("Claim" or "None") which shows whether the word is a "Claims word". The "Claim word" is a set of pre-trained words that are unique in claim sentences. These words are extracted based on their TF-IDF scores. It will be discussed in the next section.

Topic words This is a binary indicator ("Topic" or "None") which shows whether the word appears in the topics. The topic here is the same as the topic introduced in the previous section.

When given a sentence S, this sentence is considered as a sequence of words $\{W_1, W_2, ..., W_n\}$, each word in the sentence is transformed into a tuple of 5 indicators mentioned above. That is. the word W_i is now transformed into a tuple $< W_i, POS_i, Sentiment_i, Claim_i, Topic_i >$.

5.2.3 Claim Words

The idea "Claim Words" is introduced by IBM's researchers in their research paper. [16] However, in that paper, it didn't mention how to get the "Claim Words". In this thesis, we extract the set of "Claim Words" with the TF-IDF score of the words. The dataset contains two categories of sentences: claim sentence and non-claim sentence. Since the data is heavily imbalanced, the number of non-claim sentences are around 30 times of the claim. we first randomly choose 3000 non-claims sentences from the dataset. These sentences, along with all the claim sentence, are used as the new dataset to extract the "Claim words". Then, we remove the stop words from the sentence since they are general words that will be used in nearly all sentences. After that, each sentence is treated as a document. The TF-IDF algorithm is applied and the top 50 words with the highest scores are considered as "Claims Words". Finally, we repeat the same process 30 times to exclude the bias caused by using only a small part of the data. The final score of each word is the average of all 30 runs. The words, marked as "Claim words" through this process, includes "argued", "should", "believe" et al. Which is indeed some words that will be commonly used when making a claim.

5.2.4 Extended PrefixSpan Algorithm

Some sequential pattern mining algorithms will find all possible sequences from the data even if some of the patterns can be confirmed as the non-frequent patterns during the process. Several pattern growth algorithms were created to speed up this process. One of the most famous pattern growth algorithms is the Prefix-projected Sequential Pattern Mining (PrefixSpan) algorithm. This algorithm will start with the length 1 pattern, which is a single element. This algorithm will count the frequency of each pattern, remove it if it has low frequency. The remained patterns are extended by adding one more element to them. The algorithm then repeats the counting step. By doing this, the number of candidate patterns decreased as the length of pattern increase. When dealing with a large dataset, this algorithm is significantly faster than other algorithms.

Let α be a sequential pattern in sequence database S, and β be a sequence having prefix α. The support count The α-projected database, denoted as $S|_\alpha$, is the collection of postfixes of sequences in S with prefix α. [29]

In general, the PrefixSpan algorithm can be described as follows: [29]

1. Scan $S|_\alpha$ once, find the set of frequent items b such that b can be assembled to the last element of α to form a sequential pattern. Or $< b >$ can be appended to α to form a sequential pattern.
2. For each frequent item b, append it to a to form a sequential pattern α', and output α'
3. For each α', construct $S|_{\alpha'}$, and repeat.

In this thesis, we implemented and extended PrefixSpan algorithm for searching discontinuous pattern in a given sentence. As we discussed in the previous section, the sentence now is transformed into a sequence of words and each word W_i is represented as a tuple of 5 indicators $< W_i, POS_i, Sentiment_i, Claim_i, Topic_i >$. The algorithm can be described as follow:

1. **Step 1** Search and count all the indicators appeared in the sentence. Each word will be checked five times since it contains five different indicators.
2. **Step 2** Remove patterns whose number of appearance is lower than the threshold.
3. **Step 3** Record the index of the last word used. It prevents the algorithm to go backward. For example. The last item in the pattern are indicators from word W_i, the index is set to i.
4. **Step 4** Expand the existing patterns by adding one more indicator to them. The algorithm will check all words after the record index. item **Step 5** Repeat Step 2 until the maximum length is reached.

In the first iteration, the index of the first appearance of the indicator is recorded. After that, the index of the last appearance of the indicator is used instead.

In this research, the maximum length of a pattern is set to 3, the minimum length is set to 2 And the threshold is set to half of the total number of sentences. This algorithm is applied to both claim sentences and Non-claim sentences. Patterns that are frequently appeared in both datasets are removed. Figure 12 shows a procedure of the whole process. Finally, there are 28 patterns found by the algorithm. In IBM's research, [16] They mentioned that the pattern [that,Topic,Sentiment] and [IN,Topic,Sentiment] are two significant patterns of claim sentence. These patterns are also extracted by the algorithm implemented in this thesis.

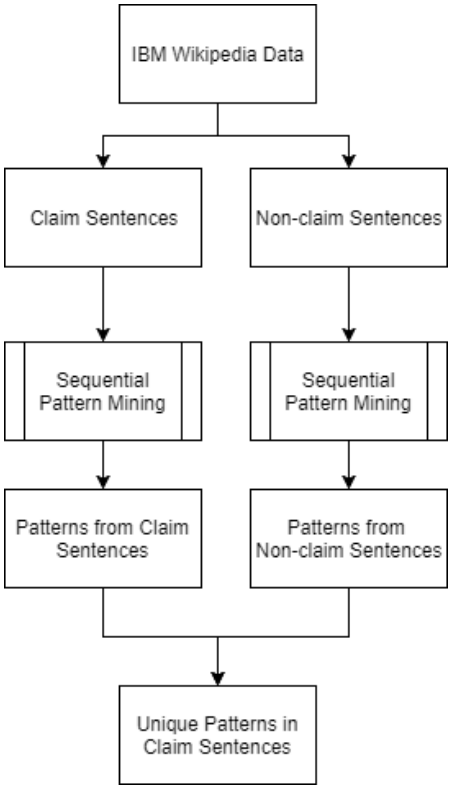


Figure 12: The procedure of extracting unique patterns of claim sentences

5.2.5 Sequential Pattern Classifier

All 28 patterns $\{P_1, P_2, ..., P_{28}\}$ are used in the sequential pattern classifier. A given sentence will be represented by a 28 dimensions vector $\{v_1, v_2, ..., v_{28}\}$. The feature v_n is binary feature (0 or 1), which shows whether the pattern P_n is appeared in sentence.

Xgboost algorithm and KNN algorithm are also used to build this category of classifiers. Figure 13 shows a basic process of this classifier. The training process is similar to the first category of classifiers mentioned in the previous sub-section. Two different data balancing strategies are also used.

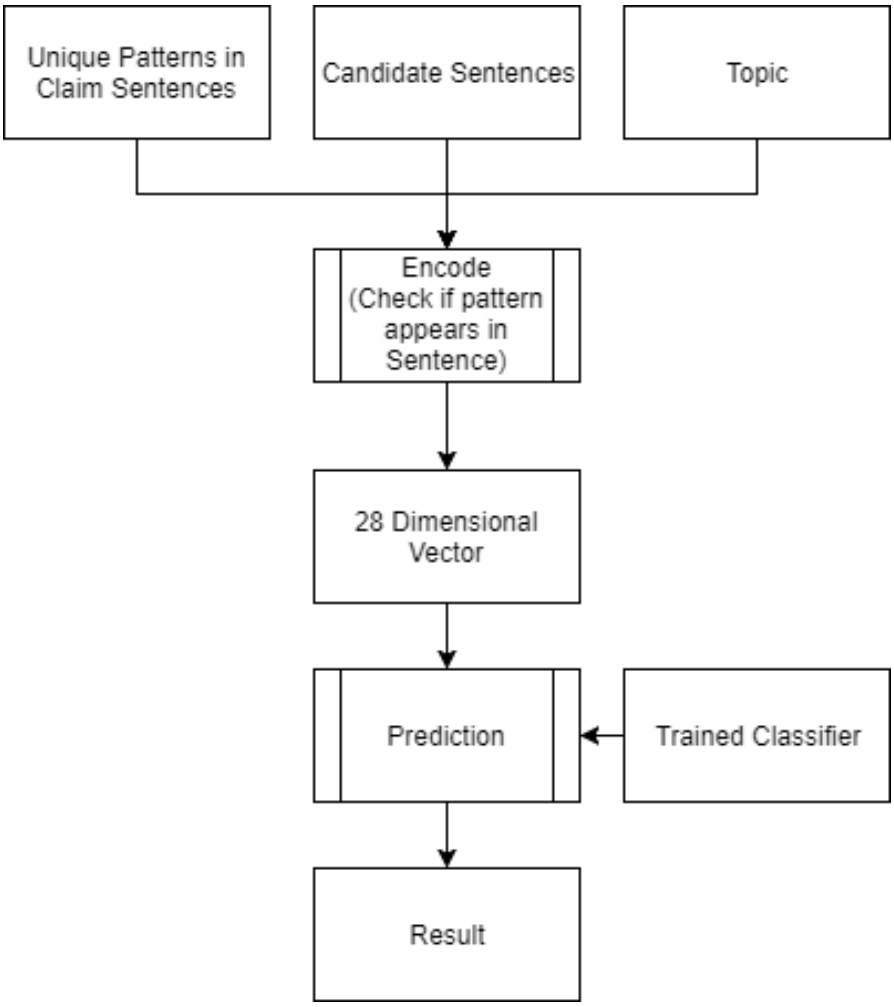


Figure 13: The procedure of the classifier using unique patterns of claim sentences

In IBM's research [16] The classifier takes the sentence component features, and sequential pattern features together to perform the prediction. However, in this research, we found that using them separately can already provide a high perception result. Using them together can easily make the classifier over fitting, which will make the classifier fail to detect any claim from given TED Talk subtitles. The experiment set up and the result will be discussed in section 7.

Following this approach, 4 different classifiers are built they are:

SPM-XGB-SMOTETomek Sequential pattern mining classifier using Xgboost algorithm and SMOTE + Tomek data balancing strategy

SPM-XGB-group Sequential pattern mining classifier using Xgboost algorithm and grouping data balancing strategy

SPM-KNN-SMOTETomek Sequential pattern mining classifier using KNN algorithm and SMOTE + Tomek data balancing strategy

SPM-KNN-group Sequential pattern mining classifier using KNN algorithm and grouping data balancing strategy

5.3 Tree Kernel SVM Classifier

Lippi et al. [19] provide an approach to detect context independent claim from the text. That is using the tree-kernel based SVM algorithm. As illustrated in section 3, there are quite some

similarities between the structure of constituency parse trees of claim sentences. If we can measure the similarity between trees, we could be able to build a boundary to group claim sentences thus extract claims from a given text. Although Lippi et al. are aiming at obtaining context independent claims, it is possible to add some additional components to the system that calculate the relatedness between sentence and a given topic and make the system only extract context dependent claims.

A Tree Kernel (TK) is designed to measure the similarity between two trees by evaluating the number of their common substructures (or fragments). [19] There are different types of tree kernel that can be used in this work. According to the definition of fragments, different tree kernels have been defined including the Sub-Tree Kernel (STK) the Sub-Set-Tree Kernel (SSTK) and the Partial-Tree Kernel (PTK). [21] STK covers all nodes from the tree, along with the trees grows from these nodes. SSTK is more general than the STK. It also includes the sub trees that don't end with terminal nodes. And finally, the PTK is the most general one. Any possible part of the original tree is included in the partial tree set. The more general the kernel is, the more time it will take to calculate.

Alessandro Moschitti [30] implemented the tree kernel algorithms with SVM-light tool kit. This research will use Alessandro's project to train the tree kernel based SVM classifier and classify the given sentences. Figure 14 shows the procedure of tree kernel based SVM classifier. When given a sentence, it is first processed by a constituency tree parser. This can be done through Stanford CoreNLP tool kit which provides a constituency tree parser. Next, the constituency tree of the sentence is passed to the tree kernel classifier which will provide the classification result. This classifier is also a binary classifier.

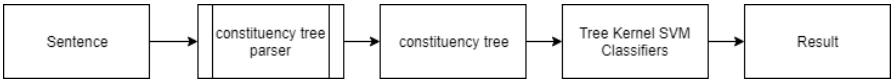


Figure 14: The procedure of tree kernel based SVM classifier

During the training process, dealing with the imbalance data is still a huge challenge. In this case, the SMOTE + Tomek strategy can't be used anymore since their inputs are not numerical. Thus, only group strategy is applied when training a tree-kernel based SVM classifier. However, during the experiment, we found that tree-kernel SVM classifier can deal with imbalanced data much better than the other two classifiers. The result will be discussed in section 7.

Following this approach, 3 different classifiers are built, they are:

ST SVM classifier using sub-tree kernel and grouping data balancing strategy.

SST SVM classifier using sub-set-tree kernel and grouping data balancing strategy.

PT SVM classifier using partial-tree kernel and grouping data balancing strategy.

6 TED Talk Claim Detection System

The core component of this claim detection system is a classifier. This classifier can indicate if an input sentence contains context dependent claims relevant to the given topic. Some additional components will be added to improve the classification result. Ideally, this classifier should be built with data from TED Talk subtitles directly to bypass the cross-domain learning problem. However, as we discussed before, there isn't such dataset exists and building one in a short period is impossible. Thus, we seek an alternative solution that is building the classifier with existing large, well-annotated dataset. The best choice would be the IBM's Wikipedia dataset since it's nearly 20 times as large as the second largest one and has been improved by researchers for two years.

In this thesis, we introduce two additional components to improve the performance of detecting sentences containing context dependent claims from TED Talk subtitles with a classifier trained on the Wikipedia dataset. sub-sentence generating component is aiming at improving the recall while topic relatedness filter is built for improving the precision.

6.1 Sub-sentence Generating Component

One of the most significant characteristics of the sentences in TED Talk subtitles is that the sentences are often complex sentences. For example, a sentence containing claim is "Not only did that cancer diagnosis change the life of our family, but that process of going back and forth with new tests, different doctors describing symptoms, discarding diseases over and over, was stressful and frustrating, especially for my aunt". The actual claim is "(cancer diagnosis) was stressful and frustrating" which is only a small part of the sentence. The rest parts of the sentence may interfere the classifier and make this sentence fail to be detected. In fact, when given this whole sentence directly to the sentences component feature classifier, it will be considered as a sentence without a claim. However, we can easily obtain several sub-sentences from a given sentence by spitting it with punctuations and the word "that". These sub-sentences are shorter and have a higher potential to only includes the claim phrase. Thus, generating sub-sentences might help in improving the recall of classifiers when extracting context dependent claims since a sentence now has higher chance to be classified a sentence containing context based claim.

The sub-sentences generate component will produce several sub-sentences from a given sentence by splitting it with punctuations and the word "that". Next, all the sub-sentences along with their original sentence will be put into a group. The classifier will check each sentence in this group. Classifiers are modified so that it can predict with the sub-sentences group. As we introduced in the previous section, the output of the classifier is a fraction indicates the probability of the given sentence containing context dependent claim. The highest probability in the group will be used as the probability score of the given sentence. Figure 15 shows a modified procedure of classifiers which takes the set of sub-sentences of a sentence as input.

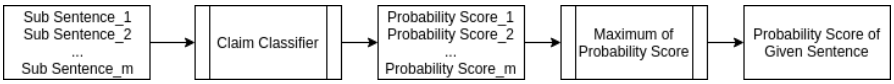


Figure 15: Modified classifier procedure which takes a set of sub-sentences as input

6.2 Topic Relatedness Filter Component

Since our goal is to extract context dependent claims, the sentences we extract should all be highly relevant to the given topic. If a sentence isn't relevant to a given topic, the probability that this sentence contains context dependent claims could be low. Besides, the sentences in TED Talk subtitles often use pronoun such as "it" or "That" as the subject of a sentence. And sentence component feature classifiers mainly considered the relatedness between the topic and the subject of the sentence. In this case, the relatedness between the subject of the sentence and the topic become less reliable. In addition, sequential pattern mining classifiers and tree-kernel support vector machine classifiers don't consider the relatedness at all during the prediction. Thus, building a filter that checks the relatedness between the topic and the whole sentence may help in excluding sentences that are not relevant to the given topics and improve the precision of the classifiers.

This thesis introduces an additional component called topic relatedness filter component. It takes as input a sentence as well as its probability that contains context dependent claim. This component then calculate the relatedness between the sentence and the given topic. The relatedness score is a fraction ranged from 0 to 1 indicate how much the sentence is relevant to the given topic. And finally, the probability score and the relatedness score will be combined. Sentences are selected based on the combined score. Only sentences whose score passed the threshold will be predicted as sentences containing context dependent claims.

In this component, we considered using 2 different way to combine two scores. Figure 16 illustrate the procedure of the cascade strategy. The sentences are first filtered by their probability score. Sentences that have less than 50% chance to contain context dependent claims are dropped in this step. Next, the remaining sentences are filtered by their topic relatedness score. Again, sentences that are not so relevant to the topic are dropped.

To find a proper threshold for the second step, we take all sentences as well as their given

topic from both the Wikipedia dataset and the persuasive essays dataset. Then, we calculate the relatedness scores between every sentence-topic pair. The average relatedness among all sentence containing context dependent claims in the Wikipedia dataset is 0.63. And the average score in persuasive essays dataset is 0.61. Thus, the threshold is set to 0.6 since our system is focusing on getting a high precision. Using this threshold, which is only slightly lower than the average level, can make the system stricter and only detect sentences that are highly possible to contain context dependent claims and increase the precision.

Figure 16: Cascade strategy

Figure 17 shows the procedure of linear combination strategy. The probability score P and relatedness score R are combined by formula $C = \frac{(P+R)}{2}$. Where C indicates the combined score. If the combined score of a sentence is less than the threshold, the sentence will not be included in the result. Instead of simply using 0.5 as the threshold, we found that finding a proper threshold is quite difficult when linear combination strategy is used. Because the probability score and relatedness score can differ a lot between 2 different TED Talks subtitle and topic pairs. During the experiment, we found that 0.7 could be a better choice of threshold.

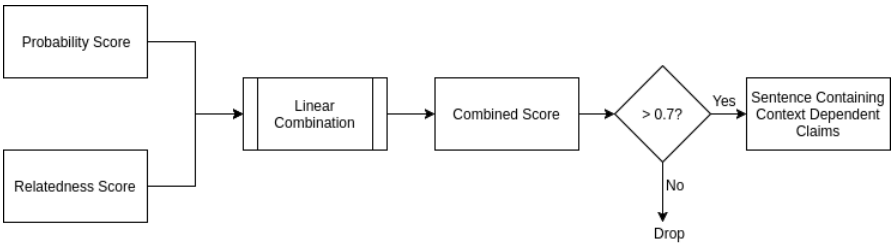


Figure 17: Linear combination strategy

The setup and result of the experiments that evaluate these two components will be discussed in the next section.

6.3 Procedure of the System

Figure 18 shows the procedure of the Ted Talk claim detection system. This system is a cascade of 4 components. First, the sentence segmentation component takes a whole TED Talk subtitle and a topic as input. This component will split the subtitle into sentences. For each sentence, the sub-sentence generating component will try to find out all possible sub-sentences from the given sentence. The output of the sub-sentence generating component is a set of sub-sentences of given sentence. Next, the claim classification component will take a set of sub-sentences as input. The modified claim classifier in this figure is introduced in section 6.1. And the output of this component will be a set of probability scores. Finally, sentences generated by the sub-sentence generating component as well as their probability scores are used as the input to the topic relatedness filter component. It will calculate the relatedness between the topic and each sentence. These relatedness scores, combined with probability scores, will give us final scores of all sentences in the given TED Talk subtitle. If the final score of a sentence is higher than 0.5, this sentence will be considered as a sentence containing context dependent claim.

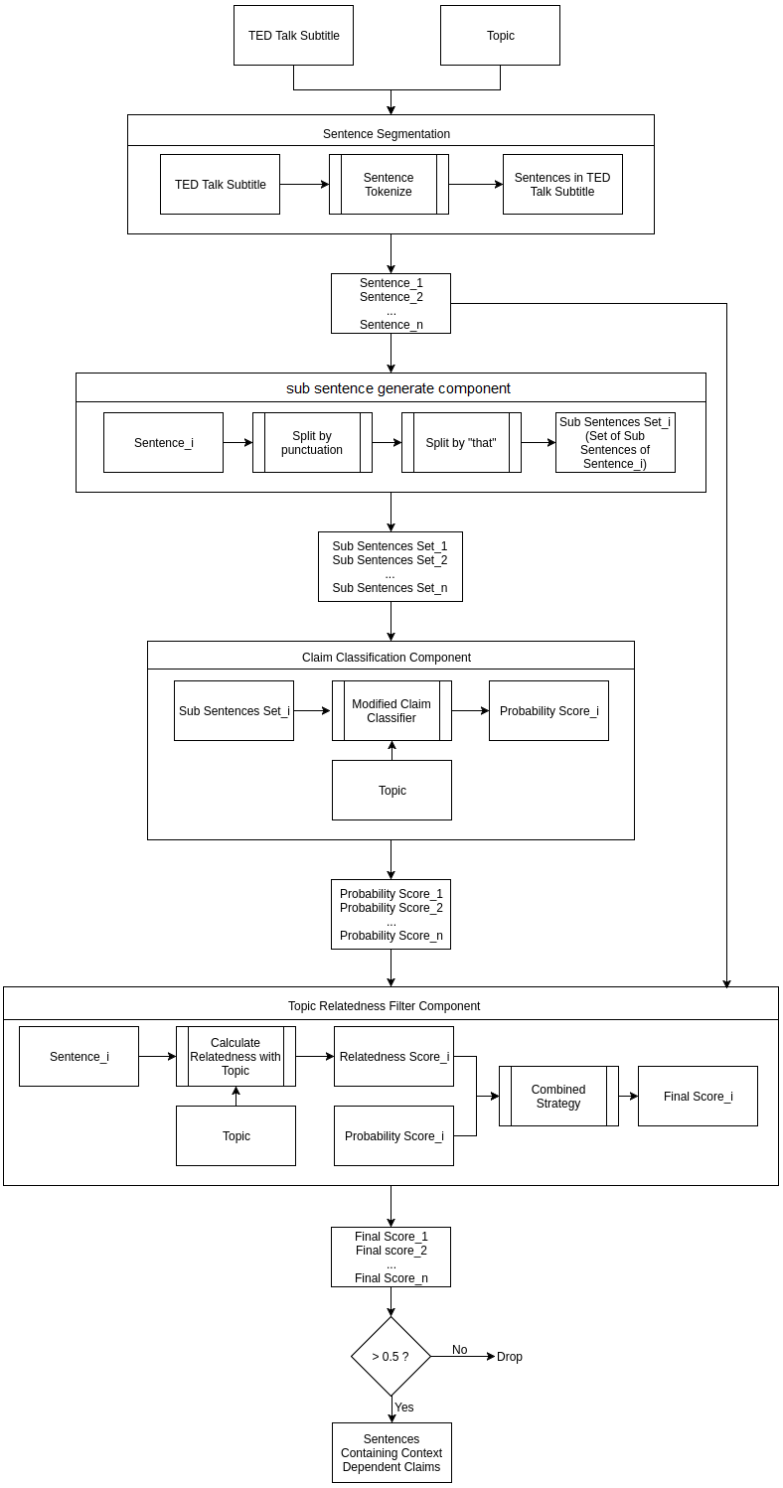


Figure 18: A procedure of extracting sentences containing claim from TED Talk subtitles

7 Experiments and Results

The goal of this research can be separated into 4 steps:

1. Check if the implemented classifiers are successful.
2. Check if the classifiers built on the Wikipedia dataset in step 1 can also work on TED Talk subtitles.
3. Provide an approach that can extract sentences containing claims from TED Talk subtitles using classifiers built with the Wikipedia dataset.
4. Validate whether the claims extracted by our system can be used to attract users to watch the recommended TED Talk.

Four different experiments will be done to evaluate each step. And three different datasets are used in this thesis. The classifiers in these three experiments are all built with the Wikipedia dataset. And one additional test will be done to validate the feasibility of using extracted claim sentences to recommend TED Talk

As we introduced in section 5, the classifiers are implemented based on three different approaches. Two classification algorithms are used, and two different data balancing methods have been applied. In total, there are 18 different classifiers implemented in this research. The first experiment is going to check the performance of all implemented classifiers with the Wikipedia dataset. An implementation will be considered as successful if the precision, recall and F1 value are high enough. Besides, the importance of having the data balancing process before training will also be verified in this experiment.

The second and third experiments are going to test the performance of the classifiers, trained in experiment 1, on datasets built with different types of texts. They will evaluate whether these classifiers suffer the cross-domain learning problem. Since our goal is to extract sentences containing context dependent claims from TED Talk subtitles, the best testing set would be a dataset built with a large number of TED Talk subtitles. However, there isn't any relevant dataset that is created using TED Talk subtitles untill now. Also, due to the challenge in the annotation process, it is nearly impossible to create a dataset as large as the existing open source dataset in a short period. An alternative solution would be finding a large, properly-annotated dataset built by articles similar to TED Talk subtitles, and use it for testing instead.

The persuasive essays dataset is the biggest one except for IBM's Wikipedia articles dataset. These essays are also focused on showing opinions. Also, the writing style of these essays is similar to TED Talk subtitles but differs a lot from the Wikipedia dataset. These characteristics make the persuasive essays dataset a good replacement of the TED Talk subtitle dataset. Thus, instead of using a dataset built with TED Talk subtitles, the persuasive essays dataset will be used as the testing set in experiment 2.

Although experiment 2 checked the cross-domain learning performance of each classifier on persuasive essays dataset, the result of experiment 2 is insufficient to validate the performance of classifiers when they are applied to TED Talk subtitles. Thus, it's still necessary to verify the performance of the classifiers and the whole system on TED Talk subtitles. Experiment 3 involves building a dataset with annotated TED Talk subtitles as well as evaluating the performance of the classifiers with this dataset. Although building a large dataset is way too time-consuming, it is possible to build a small dataset with a limited number of TED Talk subtitles. In total, 10 TED Talk subtitles are used in this experiment. Since we don't have expert knowledge about the argumentation, using crowdsourcing could be the best way to improve the reliability of the annotation. Then, the classifiers trained with the Wikipedia dataset in experiment 1 will be tested with the TED Talk dataset. If experiment 2 and 3 got similar results, we could assume that the system will have similar performance when it was applied to a much larger dataset built with TED Talk subtitles instead. Finally, the complete TED Talk claim detection system introduced in section 6 will be tested using the dataset created in this experiment.

Finally, experiment 4 is aiming at evaluate the performance of using claims as "teasing texts". We are going to compare different types of "teasing texts", evaluate their performance. In this

experiment, we check the performance of manually generated description, short texts generated by cutting edge text summarization techniques as well as claims extracted by our system. A questionnaire form is created which checks if the "teasing text" we presented can motivate users to watch the recommended TED Talk.

In summary, we are going to answer the following questions in this section:

1. Are the implementations of classifiers in this research successful?
2. Can the data balancing process improve the performance of the classifiers?
3. Can we use these classifiers trained with IBM's Wikipedia dataset to extract context dependent claims from TED Talk subtitles?
4. Can the two additional components improve the performance of the classifiers when extracting sentences containing context dependent claims from TED Talk subtitles?
5. Can we use claims extracted by our system to convince users to watch the recommended result? Are them better than manually written descriptions or texts generated by other techniques?

7.1 Experiment 1: Performance on Wikipedia Dataset

The classifiers implemented in this thesis are based on three different approaches that are introduced in previous research. These approaches are designed to extracted claims from Wikipedia articles and have been proved by researchers that perform well on IBM's Wikipedia dataset. However, both IBM and Lippi only published some general, high level descriptions of their approaches, such as what features do we need or what classifiers to use. There isn't any detail of the implementation mentioned in their papers. IBM's approaches even rely on a language parser that is not available to other researchers. Also, as we introduced in section 6, the classifiers in this thesis will be trained with the Wikipedia dataset. *Thus, the goal of this experiment is to prove that implementations of classifiers in this research are successful,* which means they can extract sentences containing context dependent claims from the Wikipedia dataset. The first experiment is going to check the performance of each classifier with the IBM's Wikipedia dataset.

In addition, since two different types of data balancing strategies are applied, this experiment will also check the performance of the data balancing strategies. The input of tree-kernel classifiers is not numerical. Therefore, the SMOTE + Tomek strategy is not used when using tree-kernel classifiers since it will not work.

7.1.1 Experiment 1.1: Evaluation of Implementations

This experiment is going to evaluate classifiers implemented in this thesis, indicate that our implementations are successful. Since approaches we followed in this thesis are only described in high-level and some of the components used in these approaches are private, we have to find our own way to implement these classifiers. Thus, it is important to know that our implementations are successful. Data balancing are not applied in all tree approaches. Thus, in this experiment, the data balancing strategies will not be applied. It makes the result of this experiment comparable with result from previous research. This experiment can answer the question 1 mentioned before.

Experiment Setup

Figure 19 showcases the procedure of this experiment. The classifiers will be trained and tested using IBM's Wikipedia dataset. This dataset contains 80000+ sentences. Only 2294 of them are sentences containing context dependent claims and the remaining are not. We divided the sentences in both categories into ten folds equally and randomly. That is, each folds in figure 19 contains around 229 sentences containing context dependent claims and 8000+ sentences without claims. In this experiment, the performance of each classifier is tested using ten folds cross

validation and is measured by precision, recall and F1 score. In each iteration, 90% of the dataset is used as the training set, and the rest 10% is used as the testing set. The ratios between claim sentences and non-claim sentences remain the same in both training and testing sets. This process will be repeated ten times, and the final result is the average among ten iterations.

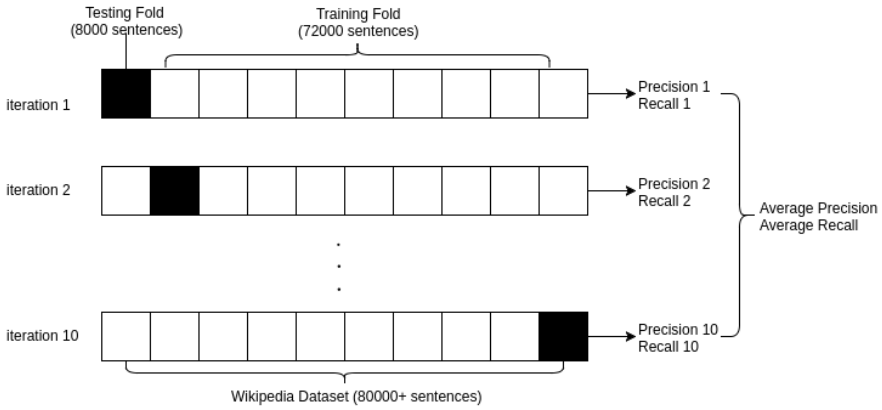


Figure 19: A procedure of this experiment

Expected Outcome

A classifier will be considered as successfully implemented if the precision, recall, and f1 scores are high enough. The precision of a classifier is the most important in this research. As we discussed before, we are using the sentences extracted by the classifier as the "teasing text". Thus, using a sentence that doesn't contain claim may have an extremely bad influence. Meanwhile, missing several claim sentences in the subtitle is acceptable as long as we can find a good claim that can motivate users to watch the recommended TED Talk.

However, there isn't a baseline of the performance of claim classifiers. Although many research has been done in this research area, there isn't any claim classifier that is focusing on getting a high precision. In IBM's research [16], the experiment setup is different in the result selection step. As we introduced in section 5, the output of the classifiers are fractions indicating the possibility of the candidate sentences containing context dependent claims. Since IBM researchers are aiming to find out the exact claim phrase of a sentence, extract sentences containing context dependent claim is only the first step of their whole system. In this step, they are aiming at extracting sentences containing context dependent claims as much as possible. In other words, they focus on the recall. Thus, the top 200 sentences with the highest possibility of containing context dependent claims will be extracted in this step. It sacrifices the precision a lot for a relatively high recall. Meanwhile, the goal of the classifiers built in this research is to extract the sentences containing context dependent claim with high precision. Since these sentences form the foundation of the recommendation system, including wrong sentences will have an adverse influence on the recommendation performance. In other words, this classifier is aiming at getting a precise result. The differences between the goals of IBM's research and this one making the result not comparable. And there is no baseline available yet for classifiers built for this purpose.

Lippi et al. reported the performance of their classifiers tested with the same experiment setup and measurement as this experiment. However, the classifiers in Lippi's research are trained and tested on persuasive essays dataset. And they are using the first version of the dataset in which only 90 essays are annotated. Also, the goal of their research is to extract context independent claims. Thus, the results of these two experiments are also not comparable.

In this case, we are using the following baseline: Since we are focusing on extracting the sentences containing context dependent claims precisely, the precision of classifier should be as high as possible. In other words, the higher the precision is, the better the classifier is. The minimum precision, in this case, should be greater than 80%. As for the recall, according to the

result published by IBM, when taking the top 50 sentences, the recall of the classifier is 40%. Thus, the minimum recall of a successfully implemented classifier should be around 40%

Result of the Experiment

Table 3 and figure 20 shows the performances of classifiers implemented in this thesis. SCF is the abbreviation of "sentence component feature classifiers", and SPM stands for "sequential pattern mining classifiers". No data balancing strategy is used in this experiment. In other words, we are using the imbalanced data to train and test classifiers directly.

According to the result, two sequential pattern mining classifiers, SVM classifier using the sub-tree kernel and sub-set-tree kernel are successfully implemented. Their precisions and recalls are higher than the baseline we set. And they all performs well on the Wikipedia dataset. SVM using the sub-tree kernel and sub-set-tree kernel may be over-fitted according to our result since the achieve extremely high scores in both precision and recall.

The SVM classifier using the partial-tree kernel failed to deal with such huge among of data (over 62000 sentences in total). SVM-toolkit throws an error message, shows that the number of identical parse nodes exceeds the current capacity. It happens when dealing with an extremely large dataset.

The recall of sentence component feature classifiers is lower than the baseline we set, which is 40%. Which means it's hard for them to detect all sentences containing context dependent claims. Using these 2 classifiers in our system may cause some serious problem since they may not able to find any sentence from a given text. In other words, the implementation of these 2 classifiers is not so successful. However, as we mentioned before, the original approach of sentence component classifiers published by IBM relies on a private parser that can only be used by IBM itself. And they both have a good precision. We think it is still possible to improve their performance by applying data balancing strategy and make the implementations successful.

Table 3: Result of classifiers using imbalanced dataset directly

		precision	recall	f1-score
Tree-kernel	ST	93.22%	94.25%	93.73%
	SST	93.17%	93.58%	93.58%
	PT	N/A	N/A	N/A
sentence component features	SCF-XGB	96.88%	27.43%	42.76%
	SCF-KNN	88.65%	15.05%	25.73%
sequential pattern mining	SPM-XGB	95.83%	61.46%	74.89%
	SPM-KNN	99.35%	68.99%	81.43%
	Baseline	80.00%	40.00%	53.33%

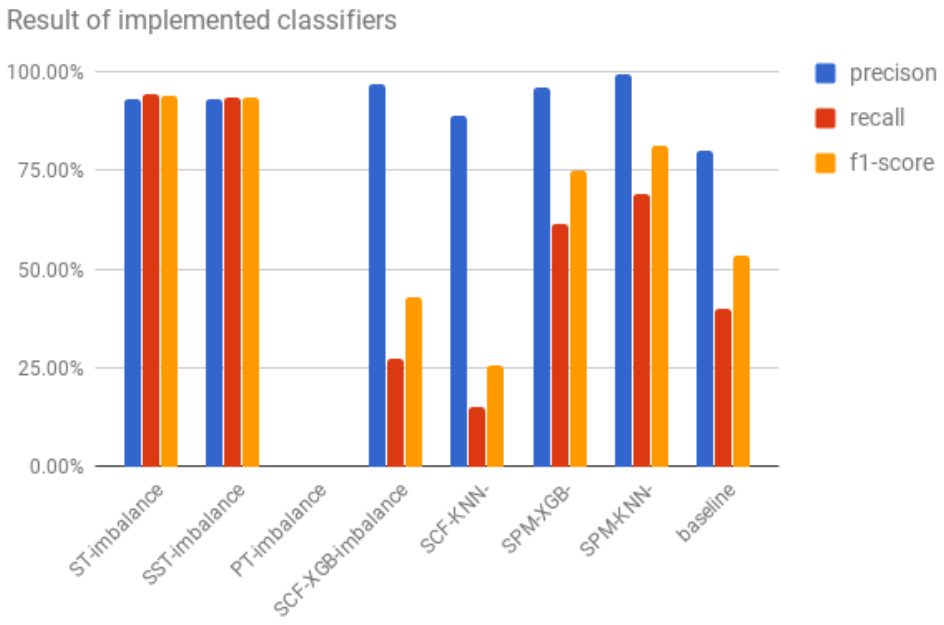


Figure 20: Result of classifiers on the Wikipedia dataset

7.1.2 Experiment 1.2: Evaluation the Data Balancing Strategy

The Wikipedia dataset is extremely imbalanced. There are over 80000 sentences in the dataset but only 2294 of them contain context dependent claims. Machine Learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced datasets. [38]. Thus, we believe that we can improve the performance of our classifiers by balancing the dataset. In this paper, two different data balancing strategies are used. This experiment is going to check if applying the data balancing strategies can improve the performance of the dataset. In other words, this experiment is going to answer the second question mentioned before.

Experiment Setup

This experiment use the same setup as the experiment 1.1. We also use 10-fold cross validation to evaluate performances of the classifiers, and the performances are measured with precision recall and f1 score. Classifiers are also trained and tested with the Wikipedia dataset. However, instead of using this dataset directly, we applied 2 different data balancing strategies to balance the training set in each iteration.

Expected Outcome

If a data balancing strategy working properly, the result of classifiers trained with balanced data should be better than classifiers trained with imbalanced data. Also, the precision and recall of a classifier trained with balanced data should also surpass the baseline we set in experiment 1.1.

Result of the Experiment

Table 4 and figure 21 shows the result of all classifiers that are tested with IBM Wikipedia data. Classifiers whose names end with "imbalance" means they are trained with the original dataset directly. If the name ends with "SMOTETomek", it indicates that the classifier is trained on a dataset balanced with SMOTE + Tomek strategy. And the classifiers whose names end with "group" are trained with dataset balanced by grouping strategy.

When using the sentence component feature approach and the sequential pattern mining approach, the performances of classifiers trained with imbalanced dataset are significantly worse than those trained with the balanced dataset. By adjusting the importance of the positive labeled data, increase the cost of predicting the positive labeled data wrong, both classification algorithms (KNN algorithm and Xgboost algorithm) can achieve good precision on both balanced and imbalanced datasets. But it sacrifices the recall and makes the classifiers stricter. It might cause some serious problem such as failing to detect any sentence from given text. By applying a data balancing strategy, we managed to increase the recall significantly. More sentence containing context dependent claims are found by our classifiers. Thus, we can say that the data balancing strategies work well in this case.

Meanwhile, imbalanced dataset seems to have little influence on tree kernel SVM classifiers. Classifiers trained with the imbalanced dataset perform only slightly worse than those trained with the balanced dataset. Thus, data balancing is not necessary for the tree kernel SVM classifiers. However, when using the grouping strategy, we actually building a group of classifiers. Each classifiers now only needs to deal with significantly smaller dataset. It allow us to get the partial-tree kernel SVM classifier work properly.

Table 4: Result of classifiers on the Wikipedia dataset

		p	r	f1
Tree Kernel	ST-group	94.27%	96.62%	95.43%
	SST-group	96.82%	96.12%	96.47%
	PT-group	88.43%	89.26%	88.84%
	ST-imbalance	93.22%	94.25%	93.73%
	SST-imbalance	93.17%	93.58%	93.58%
	PT-imbalance	N/A	N/A	N/A
Sentence Component Features	SCF-XGB-imbalance	96.88%	27.43%	42.76%
	SCF-XGB-group	90.74%	47.13%	62.04%
	SCF-XGB-SMOTETomek	89.91%	42.97%	58.16%
	SCF-KNN-imbalance	88.65%	15.05%	25.73%
	SCF-KNN-group	74.51%	82.50%	78.30%
	SCF-KNN-SMOTETomek	80.94%	81.49%	81.21%
Sequential Pattern Mining	SPM-XGB-imbalance	95.83%	61.46%	74.89%
	SPM-XGB-group	87.81%	87.17%	87.49%
	SPM-XGB-SMOTETomek	98.49%	87.08%	92.43%
	SPM-KNN-imbalance	99.35%	68.99%	81.43%
	SPM-KNN-group	98.00%	83.42%	90.12%
	SPM-KNN-SMOTETomek	98.32%	86.50%	92.03%

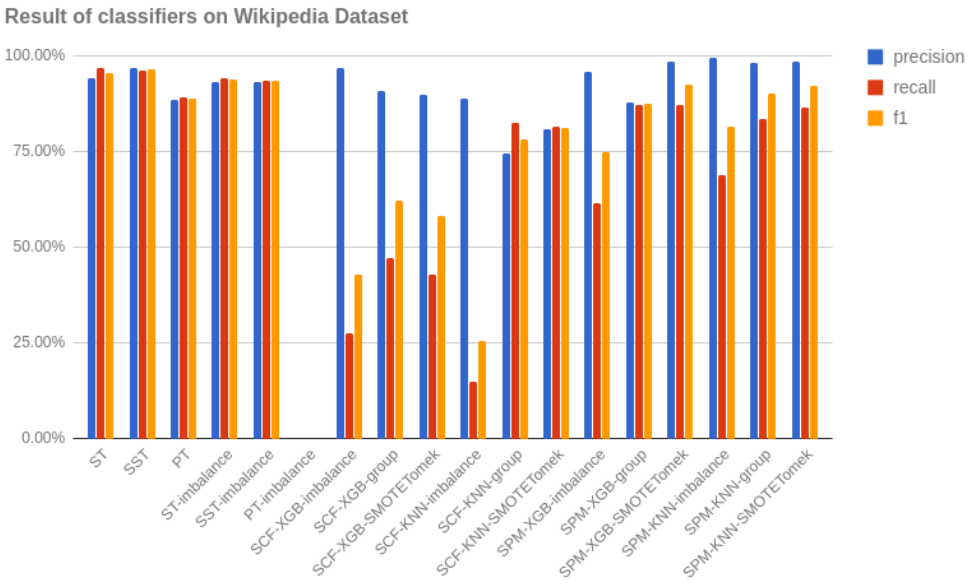


Figure 21: Result of classifiers on the Wikipedia dataset

7.1.3 Conclusion of Experiment 1

We proved in this experiment that using balanced data can improve the performance of classifiers. Both data balancing strategies work properly and can be used in our system. The tree kernel classifiers have the best performance. The precision and recall are higher than 90% most of the time. Partial tree-kernel performs worst among all tree kernel classifiers. Also, partial tree-kernel take the longest time in training process since the number of sub-trees generated by partial tree-kernel is significantly larger than those generated by sub tree-kernel and subset tree-kernel. Sequential pattern mining classifiers hold the highest precision. Although the recall is slightly lower than tree-kernel classifiers, sequential pattern mining classifiers may be the best choice to the system due to their high precision. As we discussed before, the precision is more important than recall to the whole system since including sentences that don't contain any claim in the result could have an adverse influence on the recommendation result. And sentence component features classifiers perform the worst. In this category of classifiers, those using Xgboost algorithm have better precision but much lower recall compared with those using KNN algorithm. It is acceptable since we are mainly focusing on the precision.

When data balancing strategies are applied, all implementations can be considered as successful since all precisions and recalls are higher than the baselines we set. And both data balancing strategies are able to improve the recall of classifiers significantly

7.2 Experiment 2: Performance of Cross-domain Learning

The classifiers implemented in this research, trained on Wikipedia articles data set, will be applied to texts belong to an entirely different domain, that is, the TED Talk subtitles. Experiment 1 proved that the classifiers built in this research work well on the Wikipedia dataset. Ideally, this experiment should validate the performance of each classifier on a large, properly-annotated dataset built with TED Talk subtitles so that we can check if the classifiers suffer the cross-domain learning problem.

However, there isn't such a dataset available yet. And it's nearly impossible to build such a dataset in a short time. We plan to check the performance of the classifier on 2 datasets to validate their cross-domain performance. We will first use a large properly annotated dataset built with texts that are similar with TED Talk subtitles to test the classifiers. After that, we

will build a dataset on a small number of TED Talk subtitles. If a classifier can perform well on both dataset, we could assume that it will also perform well on any given TED Talk subtitle.

This experiment is going to check the performance of the classifiers with a dataset that is sufficient in quantity, properly-annotated and built with texts that are similar to TED Talk subtitles.

Classifiers used in this experiment are also trained on the Wikipedia dataset used in experiment 1. And the testing set used in this experiment is the persuasive essays dataset built by Christian Stab et al. [36] The persuasive essays dataset is the biggest online open-source dataset except for IBM's Wikipedia dataset. Like TED Talks, persuasive essays are also focusing on sharing ideas. They are also built with argumentation structure. Also, the persuasive essays are quite different from Wikipedia articles in content. Most of the persuasive essays talk about topics that are not covered in the Wikipedia dataset. Using this dataset can also prove that the classifiers are not domain specific and can deal with topics that are not included in the training set. It's important since IBM's Wikipedia dataset only covers 52 different topics. Also, like TED Talk subtitles, most claim sentences in the persuasive essays dataset are also complex sentences, such as "Following both the point of views, the option of working or studying from home is thought to provide more benefits than drawbacks.". The phrase "Following both the point of views" is not part of the claim. These characteristics make it an excellent choice of evaluating the cross-domain learning performance. Also, the purpose of these essays is similar with TED Talk. That is to show someone's opinions. These characteristics make the persuasive essays be suitable replacements of TED Talk subtitles. However, using the persuasive dataset solely can't prove that the classifiers will surely work on TED Talk subtitles. An additional experiment is needed to check the performance of the classifiers on TED Talk subtitles. That experiment will be discussed in section 7.3.

As we discussed in section 4, the sentences labeled as "premise" in the persuasive essays dataset can also be considered as claims. This experiment will check the performance of the classifiers with datasets with and without premised sentences. If a classifier can achieve acceptable precision and recall in this experiment, this classifier also has the potential to also work on TED Talk subtitles.

7.2.1 Experiment 2.1: Cross-domain Learning Evaluation

Experiment 2.1 is going to evaluate the performance of classifiers on the persuasive essays dataset and check if they suffer the cross-domain learning problem. Classifiers tested in this experiment are same classifiers used in Experiment 1. These classifiers are also trained with the Wikipedia dataset.

Experiment setup

To compare the performance of classifiers tested with the Wikipedia dataset and the persuasive essays dataset, the setup of this experiment should be the same as experiment 1 except the training set are changed to the persuasive essays dataset. In this experiment, the testing set contains all sentences in 402 persuasive essays. And there are 6739 sentences in total. Classifiers are the same as those used in experiment 1. In other words, they are also trained on the Wikipedia dataset, and the evaluated by ten folds cross validation. The classifiers will take one sentence as input and decide whether it contains a context dependent claim. The performance of each classifier is measured with precision recall and f1 score. In this experiment, only classifiers trained with balanced data are validated since we have already proved in experiment 1 that using balanced data could improve the performance of the classifiers.

Expected Outcome

Like the last experiment, the precision of classifiers in the experiment should be as high as possible. Since these classifiers are built based on approaches that are designed to work on Wikipedia essays, it is acceptable that the precision and recall of classifiers in this experiment are slightly lower. However, if a classifier can overcome the cross-domain learning problem, the

precision and recall of this classifier should be similar to experiment 1 when tested with the persuasive essays dataset. In other words, if a classifier that performs well on predicting the Wikipedia dataset, is also able to detect sentences containing context dependent claims from persuasive essays with similar precision and recall, we could assume the classifier can deal with the cross-domain learning problem.

Result of the Experiment

Table 5 and figure 22 shows the performances of classifiers tested with the persuasive essays dataset while considering premise sentences as a special category of claim sentences. The result shows that when the classifiers, trained with the Wikipedia dataset, are applied to articles that are written in a different style such as persuasive essays, only sentences component feature classifiers still hold an acceptable result. The precisions in this group are all higher than 75% and all recalls are higher than 35%. Their performances are similar with those in experiment 1. Therefore, we can believe that these classifiers might be able to overcome the cross-domain learning problem. Tree kernel classifiers only achieve 36% in precision, which means it's hard for them to distinguish the sentences containing context dependent claim with the rest of sentences. Sequential pattern mining classifiers failed to detect any claim. All 6739 sentences in the testing set are predicted as non-claim sentences. Thus, the precision, recall and f1 score are all set to 0.0.

Also, unlike the result of experiment 1 in which the performances of KNN algorithm and Xgboost algorithm are nearly the same, the performance these two different classification algorithms differ a lot this time. Xgboost algorithm this time performs much better than the KNN algorithm. The recall of classifiers using Xgboost algorithm is almost twice as high as the recall of those using KNN algorithm. When dealing with the cross-domain learning problem, Xgboost algorithm seems to be the better choice. The two different data balancing strategies still hold similar performance. The precision, recall and f1 score of classifiers using SMOTE+Tomek strategy are almost the same as those using group strategy. Both strategies can be used in this system.

Table 5: Result of classifiers on all sentences from the persuasive essays dataset (with promise)

	precision	recall	f1
ST	17.50%	0.13%	0.26%
SST	36.00%	0.16%	0.33%
PT	16.67%	0.02%	0.04%
SCF-XGB-group	81.46%	72.53%	76.74%
SCF-XGB-SMOTETomek	81.39%	74.91%	78.01%
SCF-KNN-group	75.69%	36.94%	49.65%
SCF-KNN-SMOTETomek	77.33%	39.48%	52.28%
SPM-XGB-group/	0.00%	0.00%	0.00%
SPM-XGB-SMOTETomek	0.00%	0.00%	0.00%
SPM-KNN-group	0.00%	0.00%	0.00%
SPM-KNN-SMOTETomek	0.00%	0.00%	0.00%

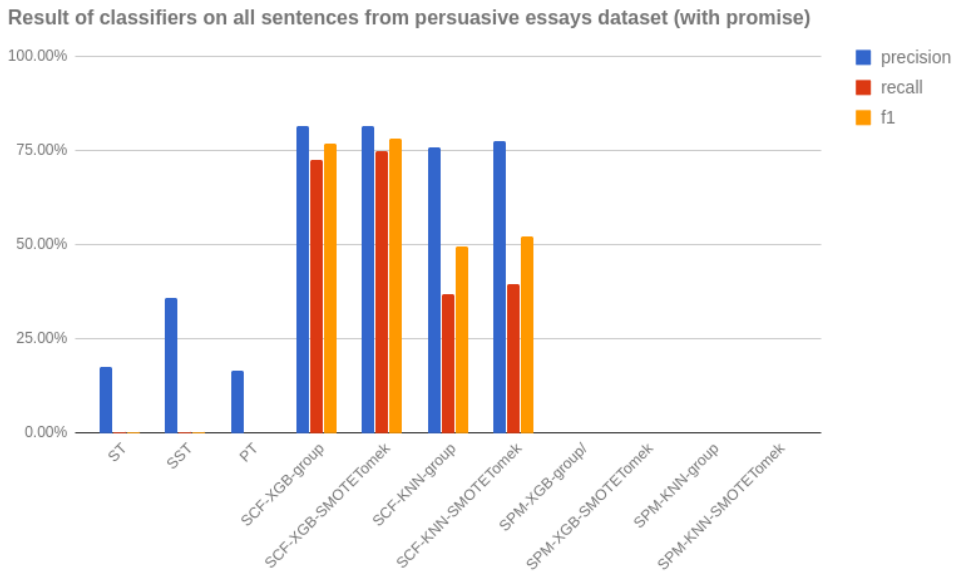


Figure 22: Result of classifiers on all sentences from the persuasive essays dataset (with promise)

Table 6 and figure 23 show the result of all classifiers applied to the same testing set. However, premise sentences are excluded in this experiment. Which means only sentences that are directly labeled as "Main Claim" or "Claim" are used as claim sentences. This time the precisions of all classifiers are significantly reduced. The results are similar to the results shown in table 5. Sentence component feature classifiers still hold the best performance, but the highest precision is reduced to 33.69%. The recall is slightly increased since there are fewer sentences that are labeled as claim sentences. It's easier for classifiers to extract more of them. The result o this experiment shows that all classifiers will consider premise sentences as sentences containing context dependent claims. Since premises are claims that support other claims, they also contain crucial information. Also, they share the same structure and properties with claim sentences. Extracting premise sentences could help us get the complete argumentation structure of a given TED Talk subtitle. Hence, although the classifiers can't distinguish the differences between premise and claim, it will not cause any problem when using in the recommendation system.

Table 6: Result of classifiers on all sentences from the persuasive essays dataset (without promise)

	precision	recall	f1
ST	2.50%	0.05%	0.09%
SST	16.00%	0.19%	0.38%
PT	0.00%	0.00%	0.00%
SCF-XGB-group	33.64%	77.97%	47.00%
SCF-XGB-SMOTETomek	33.69%	80.72%	47.54%
SCF-KNN-group	29.34%	37.28%	32.84%
SCF-KNN-SMOTETomek	31.72%	42.59%	36.36%
SPM-XGB-group/	0.00%	0.00%	0.00%
SPM-XGB-SMOTETomek	0.00%	0.00%	0.00%
SPM-KNN-group	0.00%	0.00%	0.00%
SPM-KNN-SMOTETomek	0.00%	0.00%	0.00%

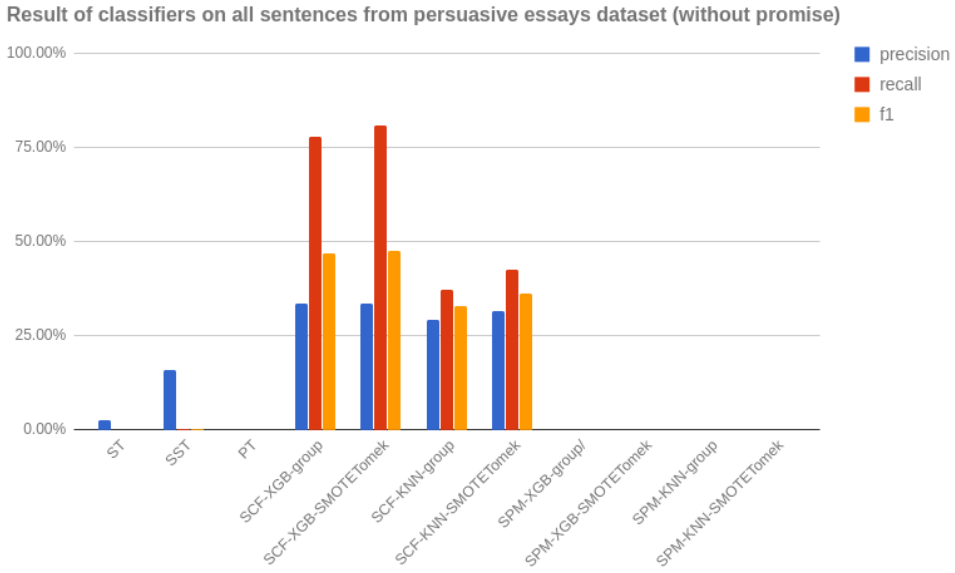


Figure 23: Result of classifiers on all sentences from the persuasive essays dataset (without promise)

According to this experiment, sentence component feature classifiers can be considered as good classifiers that can overcome the cross-domain learning problem. The Xgboost algorithm can perform better compared with KNN algorithm when dealing the cross-domain training problem. Also, both data balancing strategies are working properly. Thus, the sentence component feature classifiers using Xgboost algorithm trained with data that is balanced by either SMOTE+Tomek or group strategy are the better choice for the classifier used in this system.

7.2.2 Experiment 2.2: Performance Evaluation Under Real Use Case.

In experiment 2.1, a classifier takes a sentence as input. However, in practice, the input will be a whole article or subtitle that contains lots of sentences. The classifier should be able to extract at least one sentence from the article, otherwise, the recommendation cannot be performed. Experiment 2.1 can not evaluate this kind of performance accurately, especially the precision. For example, assume each persuasive essay have two sentences containing context dependent claims. That is 804 sentences in total. And the classifier managed to extract 402 sentences out of them. However, the classifier extracts both two claim sentences from 201 essays and failed to extract anything from the rest essays. In this case, the precision would be 100% and recall will be 50%. Both are relatively high. And the classifier might be considered as a good classifier since both precision and recall passed the baseline we set in experiment 1. But in practice, it has 50% chance fail to detect any sentence containing context dependent claim from a given text. The performance of this classifier in the real use case is unacceptable, and it cannot be used in the system. Also, although the precision is more important than recall since including sentences that don't contain context dependent claims will have an adverse influence on the recommendation result. The classifier still needs to find at least one claim from each input subtitle otherwise the recommendation cannot be performed. *Thus, experiment 2.2 is designed to evaluate the classifiers under a situation similar to the real use case.*

Experiment Setup

Figure 24 shows the procedure of this experiment. Unlike the experiment 2.1, this experiment takes a whole essay as input instead of a sentence. The precision, recall and F1 score are calculated for each article. And the overall precision and recall is the average of those get on

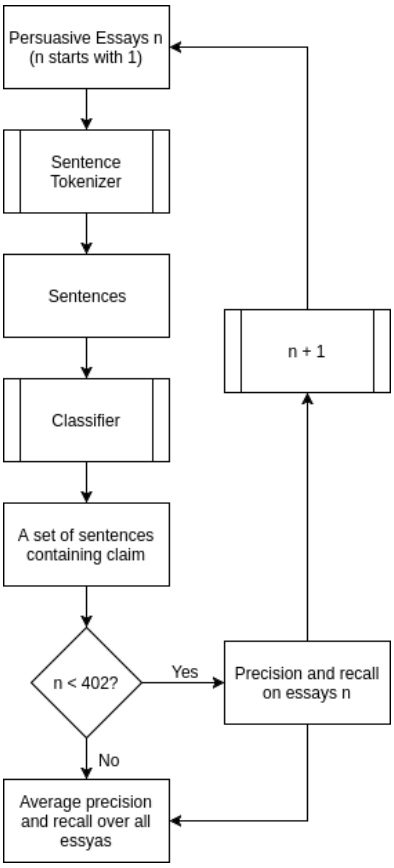


Figure 24: A procedure of this experiment

all 402 articles. The reason for doing this is to simulate the real use case because, in practice, the input will be a whole TED Talk subtitle rather than a single sentence. Thus, if the classifier fails to extract any sentence from the given article, the precision and recall are both set to 0.

There are two differences between the setup of experiment 1 and experiment 2.2. First, in experiment 1, the classifiers are trained and tested both on the Wikipedia dataset, while this experiment directly takes the classifiers that are trained in experiment 1, and tests them with the persuasive essays dataset. Second, the classifiers in experiment 2.2 take an article as input. It needs to process the article sentence by sentence. Thus, the sentence tokenizer is applied to cut the article into several sentences before it was passed to the classifier. The sentence tokenizer used in this experiment is provided by NLTK Python package which has been proved to be extremely reliable. Thus, it's safe to assume that adding the sentence tokenizer will not cause the classifier to fail at classify input sentences.

Expected Outcome

A good classifier should not only be able to extract sentences containing context dependent claims precisely but also be able to catch all of them. Although we can accept a relatively low recall to ensure high precision, the classifier should be able to extract at least one sentence out of all candidate sentences in the given text at least. Therefore, a good classifier should perform well in both experiment 2.1 and experiment 2.2.

Result of the Experiment

Table 7 and figure 25 shows the performance of each classifier on the persuasive essays dataset. The premise sentences are considered as claims. The sequential pattern mining classifiers still

have the worst performance. They fail to detect any sentence out of all 402 articles. Thus, in table 7 the precision and recall of all SPM classifiers are all 0. Tree kernel SVM classifiers also perform badly on the persuasive essays dataset. They failed to detect any claims from over 250 essays. The precision and recall also decreased a lot compared with their performance on the Wikipedia dataset. Meanwhile, the sentence component features classifiers still holds an acceptable result. The average precision is above 75% and they can extract at least one sentence from the given essays. Thus, sentences component feature classifiers can be considered as a good classifier, which can deal with the cross-domain learning problem and have a reliable performance under real use case.

Table 7: Result of classifiers on the persuasive essays dataset (with promise)

	precision	recall	f1
ST	1.45%	0.14%	0.25%
SST	1.99%	0.20%	0.35%
PT	0.00%	0.00%	0.00%
SCF-XGB-group	81.62%	73.40%	76.14%
SCF-XGB-SMOTETomek	81.49%	75.91%	77.46%
SCF-KNN-group	76.44%	48.84%	57.57%
SCF-KNN-SMOTETomek	77.00%	37.58%	48.53%
SPM-XGB-group/	0.00%	0.00%	0.00%
SPM-XGB-SMOTETomek	0.00%	0.00%	0.00%
SPM-KNN-group	0.00%	0.00%	0.00%
SPM-KNN-SMOTETomek	0.00%	0.00%	0.00%

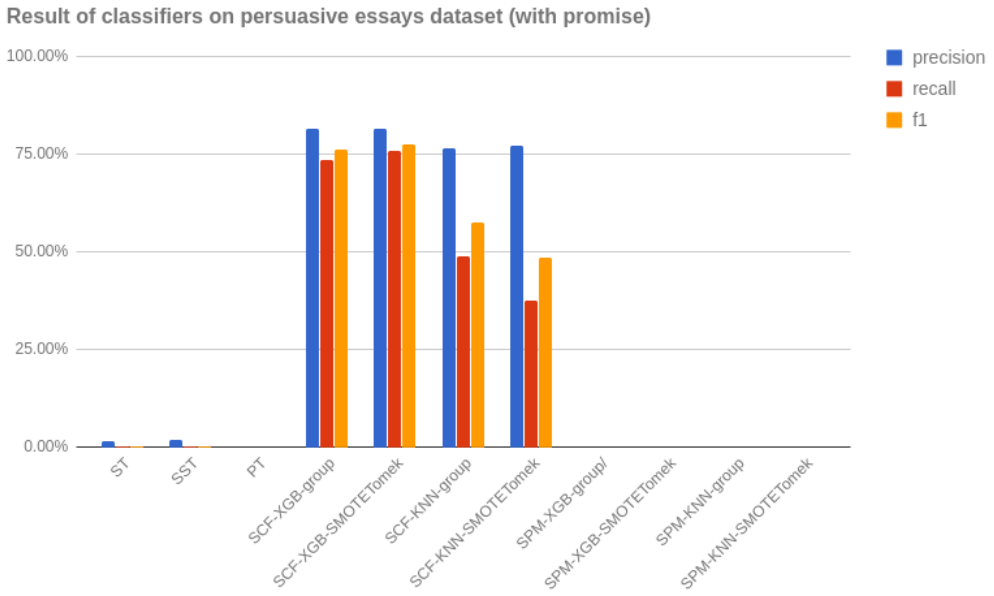


Figure 25: Result of classifiers on the persuasive essays dataset (with promise)

Table 8 and figure 26 shows the performance of each classifier when tested with the persuasive essays dataset. In this experiment, only sentences that are labeled as "Claim" or "MainClaim" in the dataset is considered as claims. The premise sentences are excluded. In this case, the precision and recall of all classifiers are significantly decreased compared with the result of each classifier shown in table 7. It means the premises will be considered as sentences containing context dependent claims by the classifiers. Meanwhile, sentence component features classifiers still hold the best performance. Sequential pattern classifiers and tree kernel SVM classifiers still

failed on detecting sentence containing context dependent claims. And this time they performed even worse. Tree kernel SVM classifiers failed on over 70% of the essays and the sequential pattern classifiers still can't detect any claim from all 402 essays. This result is similar to the result shown in table 7. Sentence component feature classifiers this time failed on around 90 articles. But they are still much better compared with the other two categories of classifiers.

Table 8: Result of classifiers on the persuasive essays dataset (without promise)

	precision	recall	f1
ST	0.25%	0.04%	0.07%
SST	0.87%	0.24%	0.37%
PT	0.00%	0.00%	0.00%
SCF-XGB-group	34.75%	78.15%	47.10%
SCF-XGB-SMOTETomek	34.84%	81.33%	47.79%
SCF-KNN-group	30.90%	33.82%	32.29%
SCF-KNN-SMOTETomek	32.16%	51.26%	37.13%
SPM-XGB-group	0.00%	0.00%	0.00%
SPM-XGB-SMOTETomek	0.00%	0.00%	0.00%
SPM-KNN-group	0.00%	0.00%	0.00%
SPM-KNN-SMOTETomek	0.00%	0.00%	0.00%

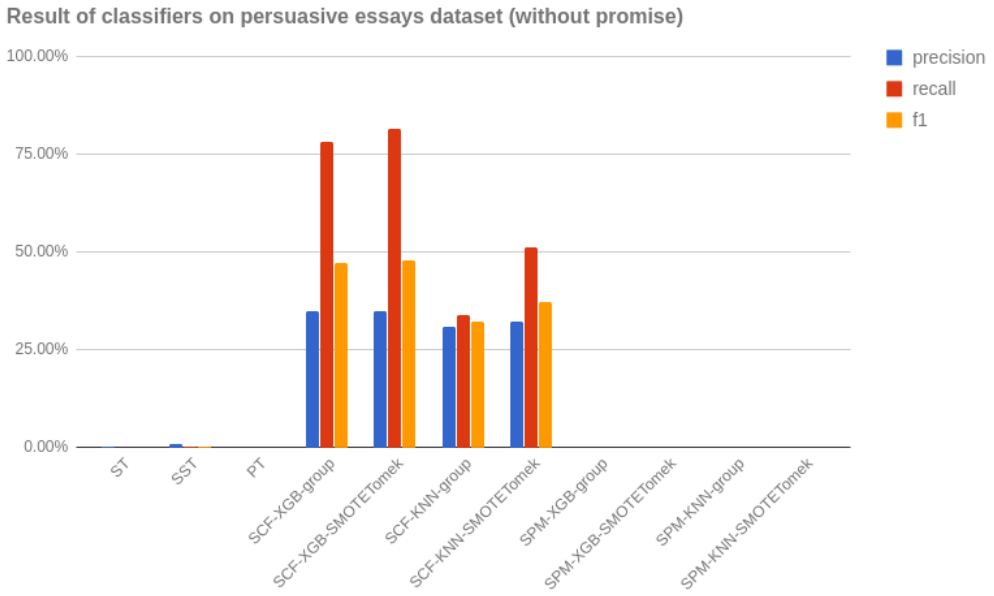


Figure 26: Result of classifiers on the persuasive essays dataset (without promise)

According to the results of experiment 2.1 and experiment 2.2, several conclusions can be drawn. First, although tree-kernel SVM classifiers and sequential pattern classifiers perform well when testing on the Wikipedia dataset, they do suffer the cross-domain learning problem and cannot be used to TED Talk subtitles. Meanwhile, sentence component feature classifiers perform well on both the Wikipedia dataset and the persuasive essays dataset. These classifiers can overcome the cross-domain learning problem. When building the TED Talk claim detection system, sentence component feature classifiers are the better choice. Second, the overall performance of classifiers built with Xgboost algorithm performs better when dealing with the cross-domain learning problem.

7.2.3 Experiment 2.3: Performance of Additional Components in TED Talk Claim Detection System

Since sentences in persuasive essays are similar with those in TED Talk subtitles, it is reasonable to also validate the performance of the two additional components in the system that were introduced in section 6. They are the sub-sentence generating component and topic relatedness filter. If the classifier performs better after adding the sub-sentences generator component on the persuasive essays dataset, we can say that this component might be able to improve the performance on TED Talk subtitles dataset. Similarly, if the classifiers can perform better when topic relatedness filter is applied, we could assume that this component will also benefit the system when dealing with TED Talk subtitles due to the similarity between persuasive essays and TED Talk subtitles.

Experiment Setup

This experiment will check the impact of the sub-sentence generating component and how can it improve the performance of the whole system. The setup of this experiment is similar to experiment 2.2 to check the performance under the situation that simulates the real use case. Only sentence component feature classifiers are used since the other two categories of classifiers are confirmed that cannot overcome the cross-domain learning problem.

This experiment compares the performances of 3 groups of classifiers. In each group, four different classifiers are included. They are sentence component feature classifiers with the different combination of classification algorithms and data balancing strategies. The classifiers in group 1 don't have any additional component. Meanwhile, classifiers in group 2 are combined with the sub-sentence generating component and classifiers in group 3 will use the topic relatedness filter to grind the result. The results are also measured with precision, recall and F1 score.

Expected Outcome

If the component can indeed improve the performance of the system. Classifiers in group 1 will have the worst performance. Classifiers in group 2 will have higher recall and classifiers in group 3 will be more precise.

Result of the Experiment

Table 9 and figure 27 shows the result of classifiers in group 1 and group 2. Similarly, the classifiers are built on the Wikipedia dataset and tested on the persuasive essays dataset. According to the result, the sub-sentence generating component has nearly no effect when the classifier is using Xgboost algorithm. However, it doubles the recall of classifiers that use KNN algorith. This component can help classifiers with KNN algorithm to extract more sentences containing context dependent claims from given texts and improve the recall significantly. Therefore, using a sub-sentence generating component might be able to improve the performance of the TED Talk claim extraction system.

Table 9: result of sub-sentence generator

		precision	recall	f1
No Sub Sentence	XGB-group	81.62%	73.40%	76.14%
	XGB-SMOTETomek	81.49%	75.91%	77.46%
	KNN-group	76.44%	48.84%	57.57%
	KNN-SMOTETomek	77.00%	37.58%	48.53%
Split into Sub Sentences	XGB-group	81.48%	73.56%	76.25%
	XGB-SMOTETomek	81.88%	75.91%	78.75%
	KNN-group	81.55%	86.12%	83.16%
	KNN-SMOTETomek	81.60%	82.48%	80.29%

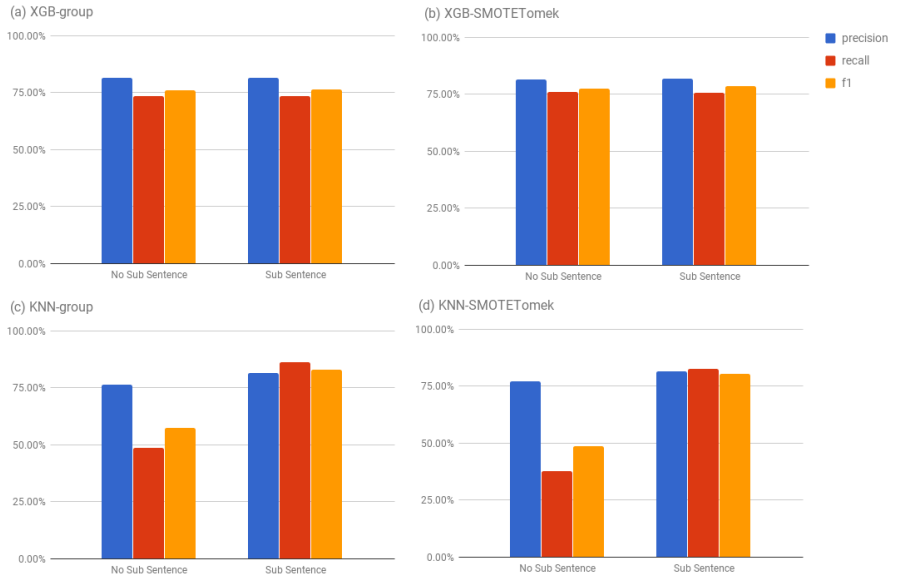


Figure 27: Result of classifiers with and without sub-sentence generating component

Table 10 and figure 28 shows the result of classifiers in group 1 and group 3. In this experiment, this additional component using cascade strategy only makes the result worse. The precision, recall and F1 score of all four classifiers are slightly lower. The decrease in the recall is the most significant since this filter will remove all sentences that are considered as "not relevant" to the given topic. In other words, all sentences with relatedness score less than 0.5 are removed. It will doubtlessly dismiss some sentences containing context dependent claims and will make the classifier more strict, harder to detect all sentences containing context dependent claims from given text. Meanwhile, when using the linear combination strategy, the system could properly weight both scores. When predicting a sentence, both probability score and relatedness score are considered properly. According to the result, when using linear combination strategy, this component managed to slightly improve the performance of the classifiers. The classifiers now have better precision and recall. Especially classifiers using xgboost algorithm.

Table 10: result of topic relatedness filter

		precision	recall	f1
No Filter	XGB-group	81.62%	73.40%	76.14%
	XGB-SMOTETomek	81.49%	75.91%	77.46%
	KNN-group	76.44%	48.84%	57.57%
	KNN-SMOTETomek	77.00%	37.58%	48.53%
Cascade	XGB-group	78.45%	59.05%	65.01%
	XGB-SMOTETomek	78.48%	60.91%	66.35%
	KNN-group	68.84%	23.94%	33.58%
	KNN-SMOTETomek	72.74%	28.22%	38.80%
Linear Combination	XGB-group	81.61%	87.07%	83.61%
	XGB-SMOTETomek	81.80%	80.12%	79.92%
	KNN-group	80.42%	51.18%	60.52%
	KNN-SMOTETomek	77.34%	38.18%	49.27%

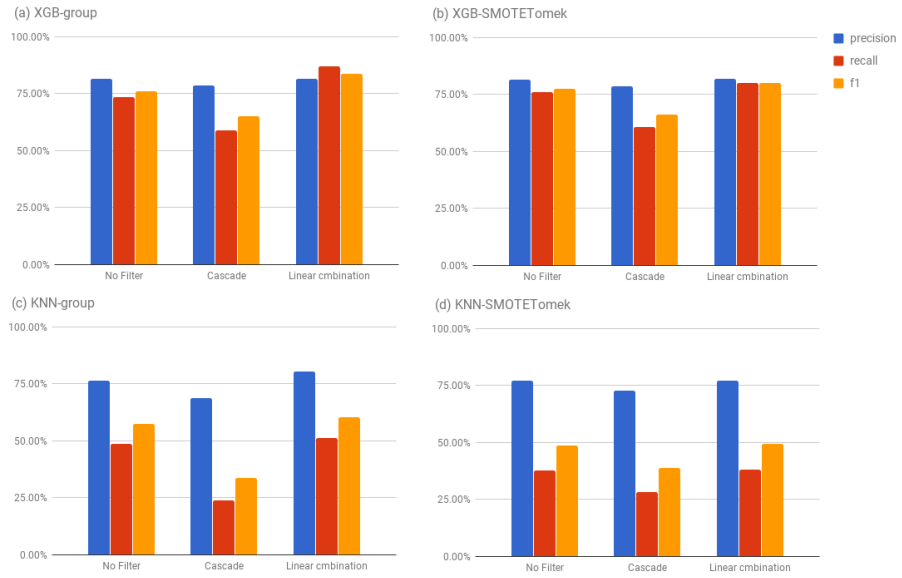


Figure 28: result of sub-sentence generator

According to this experiment, it seems that the topic relatedness filter component using cascade strategy is useless since it only makes things worse. However, we can't draw any conclusion yet since it is only tested with the persuasive essays dataset. As we introduced before, the claims in persuasive essays are context independent claims, which means claims from an essay are not necessarily related to the topic. Besides, there are still some differences between persuasive essays and TED Talk subtitles. For example, although differs a lot from Wikipedia articles, the persuasive essays still use the written language, while the TED Talks use the spoken language. These are the reasons why the topic relatedness filter component doesn't work on this dataset. Thus, the performance of these two additional components will also be checked in the next Experiment which uses a small TED Talk subtitles dataset as the testing set. We should consider results from both experiments and compare them carefully before we draw any conclusion about these two additional components.

7.2.4 Conclusion of Experiment 2

In this research, we found that only sentence component feature classifiers have the potential to overcome the cross-domain learning problem. Only classifiers following this approach are able to get an acceptable performance on the persuasive essays dataset. Tree kernel SVM classifiers and Sequential pattern mining classifiers can be considered as over-fitted. It is not a surprise that these two kinds of classifiers failed to work on the persuasive essays dataset since they are mainly capturing the unique grammars used in claims. And the way people making claims in Wikipedia articles and persuasive essays are quite different.

Sub-sentence generating component can improve the classifiers' performance on persuasive essays. It can increase the recall significantly, especially when combined with classifiers using KNN classification algorithm. Topic relatedness filter component using linear combination strategy can improve both precision and recall slightly. It may also help our system detect sentence containing context dependent claims from TED Talk subtitles. And the Topic relatedness filter component using cascade strategy is not working. However, we can not draw a solid conclusion by now since we only tested them on the persuasive essays dataset. These essays are similar but still have some differences between TED Talk subtitles. Thus we need the third experiment which tests the performance of classifiers on a small TED Talk subtitle dataset.

7.3 Experiment 3: Performance on TED Talk Subtitles

The goal of this thesis is to extract sentences containing context dependent claims from TED Talk subtitles. Thus, it is necessary to check the performance of the classifier on actual TED Talk subtitles. There are two challenges in this experiment. First, there isn't any dataset available that is built with TED Talk subtitles. The only way to evaluate the system is to create a dataset on TED Talk subtitles ourselves. In other words, we need to annotate TED Talk subtitles, manually extract claims from them and build a dataset with the human annotation result. However, due to the complexity of the annotating process, it is not possible to annotate a huge amount of TED Talk subtitles.

In this experiment, only 10 TED Talk subtitles will be annotated via crowdsourcing. Such a small dataset could make the result of this experiment not highly reliable. Using the result of this experiment solely also lack the power of proving that the classifiers are surely working on TED Talk subtitles. However, by combining the result of experiment 2 and 3, we could draw some more reliable conclusion. In experiment 2 the classifiers perform well on a large, properly labeled dataset built on essays that are similar to TED Talk subtitles. And this experiment will evaluate the performance on a small dataset that built on the TED Talk subtitles. If the result of this experiment is consistent with experiment 2.1and 2.2, that is, the precision and recall of the classifier are relatively high when tested on both the persuasive essays dataset and this small TED Talk dataset, we could assume this classifier can extract sentences containing context dependent claims from TED Talk subtitles. Also, the experiment 2.1 and 2.2 proved that only the sentence component feature classifiers have the potential to extract claims from a different type of texts. Thus, only sentence component feature classifiers will be used in this experiment.

This experiment, along with experiment 2, is going to answer the question 3 and 4 mentioned before.

7.3.1 Experiment 3.1: Building a TED Talk Subtitle Dataset

The first step of the Experiment will be building a TED Talk subtitle dataset. IBM's researchers [1] provided a guideline of how to extract claims from a given text manually. There will be several annotators participate in the annotation work. Each of them first annotates part of the articles. After that, annotators will hand their result to another annotator to do the cross-examination. This time, the annotator will check other annotators' work, and decide if they agree with the result or not. In other words, an article will be first annotated by one annotator, and the result will be reviewed by all other annotators. Only sentences that have less disagreement are used. Also, reviewing others work will be much easier than annotate a whole article, which could significantly speed up the whole process. However, this requires the annotators to have expert knowledge about the argumentation.

Since it's hard to find someone who has the required expert knowledge, and we don't have such knowledge either, following the guideline published by IBM would be impossible. One of the most popular alternative solutions would be relying on the crowdsourcing techniques. That is, having much more participators regardless of whether they have professional knowledge. And use the result that is agreed by most of the participators.

Also, the purpose of this annotation work is different from IBM's work and Christian Stab's work. In those research, researchers are aiming at extracting the structure of the argumentation hidden inside the article. Annotators are going to extract claims, evidence as well as the relations among them. And each claim they extract comes with clear boundary indicate which part of the sentence is a claim. However, this annotation work only focuses on extracting sentences containing context dependent claims. Unlike the Wikipedia dataset and the persuasive essays dataset where a claim might not be a full sentence, we use the whole sentences taking directly from the TED Talk subtitles to build this dataset. In this case, the goal of the annotation is much more straightforward and clear compared with other research. Makes it possible to rely solely on crowdsourcing and without expert knowledge.

Experiment Setup

In total, 10 TED Talk subtitles are selected to build the dataset. They are selected under 10 different topics. 3 of them are included in the Wikipedia dataset. They are "climate change", "gender equality" and "trade and aid". Other topics such as "copyright" and "big data" have never been mentioned in the Wikipedia dataset. These uncovered topics will be able to detect whether the system is topic-specific. Since the Wikipedia dataset only contains 52 topics, and there is no doubt that TED Talk will cover much more topics, it is important to prove that our system can work properly when given a topic that is not included in the training set.

This work provides a detailed and clear guideline, including the definitions of "topic", "context dependent claim" and "sentence containing context dependent claim". Also, there is an introduction of the expected format of the outcome. And finally, we provided some examples taken from the Wikipedia dataset that illustrate what kind of sentences should be extracted from the subtitle. These examples could help the participators understand the goal of this task better.

Before participators start annotating, they were required to read through the guideline document carefully. Then, each of them will annotate four articles. Usually, it would take about an hour for participators to finish the annotation.

Each annotator will annotate three TED Talk subtitles. The rest one is a persuasive essay that functions as the quality control text. The quality control texts are essays taken directly from the persuasive essays dataset. They have already been properly-annotated by other researchers. We can easily measure the performance of an annotator by comparing the annotation result of the essay to the annotation result given by the dataset. Also, persuasive essays are much shorter and more straightforward than Wikipedia articles. Annotators will not spend too much time on the quality control texts. These articles can help us get rid of free writers and improve the quality of the crowdsourcing annotation result in the evaluation step.

The evaluation of annotation result can be divided into 2 steps. First, a "quality score" is measured by the precision, recall and f1 score of the annotator's result on quality control texts. Since the dataset built in previous research all relies on expert knowledge rather than crowdsourcing, there is no baseline set by previous research. And we have no idea about the participators 'knowledge about argumentation. In this case, we take the average scores of all participators. The result from the participator who gets significantly worse result than others will be excluded. We fit a normal distribution on precisions, recalls and f1 scores from all annotators. Since the poor quality of result from annotator will harm the dataset a lot, we only use annotators whose score is between the interval $[-\delta, +\inf]$

The second step is to check the "agreement score" between annotators. In this case, the Cohen's kappa coefficient will be used to calculate the agreement scores between annotators as well as the overall agreement score of this annotation task. Figure 29 shows an example of calculating the Cohen's kappa coefficient. For each TED Talk subtitle, the kappa coefficient is calculated between the result from every pair of the annotators. For example, if a subtitle is annotated by three annotators $\{A_1, A_2, A_3\}$, there will be three kappa coefficient calculated. They are $\{kappa_{12}, kappa_{23}, kappa_{13}\}$. And the overall kappa coefficient of this TED Talk subtitles is the average among all tree kappa coefficient values.

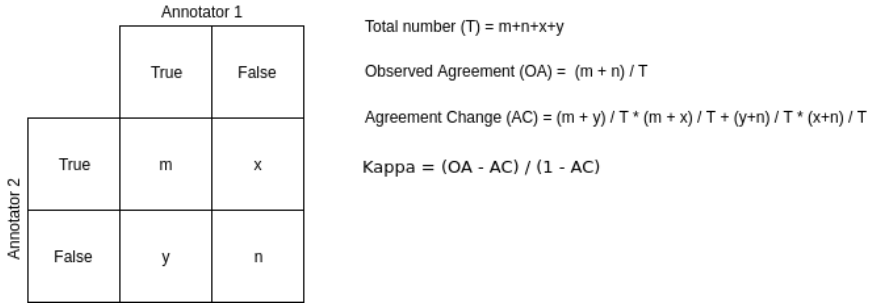


Figure 29: An example of calculating the Cohen's kappa coefficient

Expected Outcome

If the crowdsourcing annotation result is reliable, the average agreement score among all participants should be higher than the baseline. According to IBM's research, [1] The baseline agreement score is 0.39. Our goal is to reach an agreement score equal or higher than this number. However, due to the fact that participators may lack the expert knowledge, a slightly lower agreement score is also acceptable.

Result of the Experiment

We first check the performances of all annotators on the quality control texts. In this experiment, 10 TED Talk subtitles in total are annotated. And 11 annotators participated in this task. Table 11 and figure 30 show the annotation result as well as the statistical analysis of it. The average precision of all annotators is 81.95% and the average recall is 52.16%. According to the result, the result of annotation 3-2 is significantly worse than the others. The precision, recall and f1 scores of annotation 3-2 are lower than the threshold. The risk of including this annotation result is unacceptable. Thus, this annotation result will not be considered. Meanwhile, the rest annotation results successfully surpass the threshold in at least 2 measurements. Annotation 4 and annotation 9 have precisions that are slightly lower than the threshold, but their recalls and f1 scores are high enough. Therefore, they can be used to build the dataset. The TED Talks subtitle dataset will be built with the rest 10 annotation results.

Table 11: Result and Statistic Analysis of Annotation Result

	precision	recall	f1
annotation_1	85.71%	50.00%	63.16%
annotation_2	80.00%	50.00%	61.54%
annotation_3	80.00%	57.14%	66.67%
annotation_3-2	66.67%	28.57%	40.00%
annotation_4	70.00%	58.33%	63.64%
annotation_5	100.00%	50.00%	66.67%
annotation_6	85.71%	50.00%	63.16%
annotation_7	80.00%	50.00%	61.54%
annotation_8	83.33%	71.43%	76.92%
annotation_9	70.00%	58.33%	63.64%
annotation_10	100.00%	50.00%	66.67%
mean	81.95%	52.16%	63.05%
std	11.01%	10.28%	8.77%
threshold	70.94%	41.88%	54.29%
min	66.67%	28.57%	40.00%
max	100.00%	71.43%	76.92%

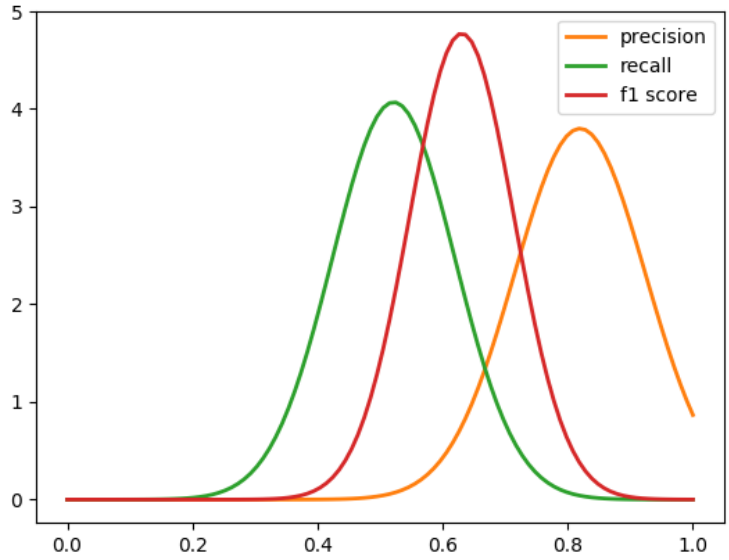


Figure 30: Distribution of precision, recall and f1 score of the annotation result

Like the dataset published by IBM, we used Cohen's kappa coefficient to calculate the agreement score among all annotator pairs. The average of agreement scores we get is 35.29%, which is slightly worse than the baseline (39%) published by IBM. As we introduced before, IBM's use professional annotators with expert knowledge of argumentation, while we rely solely on crowdsourcing. The annotators participate in this job are mainly students, and some of them didn't know anything about what the claim is before reading the instructions we provided. Thus, a slightly lower agreement score is acceptable. Also, it proves that our crowdsourcing annotation task is properly designed. By providing the guideline which includes the definition of terms used in this task as well as some properly selected examples, annotators understood what we are looking for exactly and did the same job looking for the same kind of sentence. In other words, the annotators who participate this task are reliable.

In this annotation task, each TED Talk subtitle is annotated by 3 different annotators. According to the performance of annotators on the quality control texts, the average precision is 81.95%. The change that a sentence is miss-labeled by 2 annotators is around 4%. Thus, if a sentence is marked as a sentence containing context dependent claim by at least 2 annotators, this sentence will be considered as a sentence containing context dependent claim. There are 96 sentences found by annotators in total among all 10 TED Talk subtitles. And each TED Talk has at least 3 sentences containing context dependent claims. Considering that the average recall of annotators on quality control texts is only 52.18%. Annotators might miss some sentences from a given TED Talk subtitle. The reason of having a relatively low recall is that annotators are strict with relatedness.

Meanwhile, the precision indicates that the result is precise, which means all sentences extracted by annotators are indeed sentences containing context dependent claims. This experiment proved that crowdsourcing can provide an acceptable result, but the annotators cannot extract all sentences containing context dependent claims from the TED Talk subtitles. The dataset built with these annotation results is acceptable but not highly reliable.

After all annotating job has been finished, a dataset will be built with the annotation result. The next step is to test the overall performance of the classifiers and the whole system and with two additional component. Experiment 2 has proved the tree-kernel SVM classifiers, and sequential pattern classifiers are unlikely to have good performances on the TED Talk subtitle dataset. Thus, only sentence component feature classifiers will be checked in the following

experiments.

7.3.2 Experiment 3.2: Classifier Performance Evaluation

Experiment Setup

In order to make the result comparable to results on persuasive essays data and Wikipedia articles data. This experiment follows the same structure used in experiment 1 and experiment 2.1. That is, a dataset contains all sentences from 10 annotated TED Talk subtitles will be used as the testing set. The classifiers, trained with the Wikipedia dataset in experiment 1, will be applied to predict whether the input sentence from TED Talk dataset contains a context dependent claim. Thus, the result, measure with precision recall an f1 score, can be compared with results in experiment 1 and experiment 2.1. If a classifier in this experiment got a relatively high precision, recall and f1 score, which means the results are consistent with all 3 experiments, we could further confirm that this classifier can be used to extract sentences contain claims from TED Talk subtitles.

Expected Outcome

Ideally, the classifier should be able to achieve similar precision and recall compared with the result of experiment 1 and 2.1. If the results of these three experiments are consistent, we can give an affirmative answer to question 3.

Result of the Experiment

Table 12 and figure 31 shows the result of classifiers tested on TED Talk subtitles dataset built in the previous step. According to the result, the precisions decreased a lot compared with the result on the Wikipedia dataset and the persuasive essays dataset. The highest precision is only 15.29%. Also, the recalls have also decreased a lot, which means the classifier will fail to detect some sentences containing context dependent claims.

After checking the output of our classifiers carefully. We found two main problems that caused the decreasing of precision and recall. First, the sentences marked as sentences containing context dependent claims are not directly supporting or against the given topic. For example, in a TED Talk which is talking about "big data", the sentence "Privacy was the central challenge in a small data era." is included in the output. This sentence can be considered as a premise, and as we discussed in experiment 2, our classifiers will include premises in their result. However, during the annotation process, only claims that are directly supporting or against the given topic are extracted. Meanwhile, unlike Wikipedia articles and persuasive essays that are highly focused on the topic, a TED Talks often use a huge number of premises to support its main claim. And these premises are focused on some detailed aspect of the main topic. For example, in a TED Talk whose main claim is "Aid is a bad instrument to Africa", the speaker talked a lot about how media are misleading the western countries' view of Africa's economic dilemma. This is the most common situation in TED Talks.

Second, we found that the classifier will fail to detect some long, complex sentences. For example, the sentence "The concern is really that we will build machines that are so much more competent than we are that the slightest divergence between their goals and our own could destroy us." in a TED Talk which is talking about "AI losing control" is not included in the outputs of our classifiers. Because the structure of this sentence is quite complicated, it's hard to find the proper subject of this sentence.

The way to improve the performance of the system is to simplify the sentences and filter them by their relatedness scores with topic. Thus, the two additional components, namely sub-sentence generating component and topic relatedness filter component should be able to improve the performance of the whole system.

Table 12: Classifiers tested on TED Talk subtitle Dataset

	precision	recall	f1
XGB-group	14.79%	39.58%	21.53%
XGB-SMOTETomek	15.29%	38.54%	21.89%
KNN-group	9.75%	44.79%	16.01%
KNN-SMOTETomek	12.01%	41.67%	18.65%

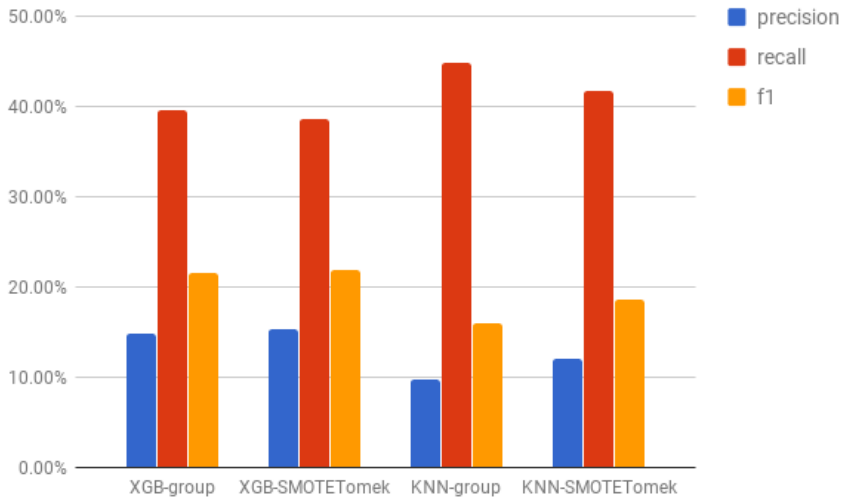


Figure 31: Classifiers tested on TED Talk subtitle Dataset

7.3.3 Experiment 3.3: Classifier Performance Evaluation in Real Use Case

Also, like experiment 2.2, classifiers should be able to extract at least one sentence containing a context dependent claim from the given TED Talk subtitles. This experiment is going to check the performance of classifiers under the situation that is similar to the real use case. In this experiment, the TED Talks subtitles will be used as the input instead of single sentences. The precision and recall of this experiment will be the average value measured among all 10 subtitles. If a classifier fails to detect any sentence in most of the TED Talk subtitles, it shouldn't be used in practice.

Experiment Setup

The setup of this experiment is the same as experiment 2.2. Classifiers take a whole TED Talk subtitle as input. For each subtitle, precision, recall and F1 score are used to measure the performance of the classifier applied to it. The performance of a classifier is measured by the average of it precision, recall and F1 score among all 10 subtitles

Expected Outcome

If a classifier can perform similarly on this small TED Talk subtitle dataset and the persuasive essays used in experiment 2, we can assume this classifier is able to perform well on TED Talk subtitles generally. Compared with persuasive essays, TED Talk subtitles are much longer, which means there are more non-argument content. Also, the sentences are more complex compare with sentences in persuasive essays and Wikipedia articles. Extracting sentences containing context dependent claims is harder. Thus, we assume that the performance of classifiers on TED Talk subtitle might be worse than the performance on persuasive essays.

Result of the Experiment

Table 13 and figure 32 shows the performance of classifiers under real use case. The input of the classifiers is a whole TED Talk subtitle rather than a single sentence. The result is similar to the result in experiment 3.1. All classifiers manage to detect at least one sentence from the given TED Talk subtitle, but the precision recall and f1 score are still relatively low. The reasons are already described in experiment 3.1. We believe the 2 additional component could improve the performance of the system.

Table 13: Classifiers tested on TED Talk subtitle Dataset in Real Use Case

	precision	recall	f1
XGB-group	15.78%	35.38%	20.73%
XGB-SMOTETomek	17.66%	39.10%	22.62%
KNN-group	12.37%	38.05%	17.94%
KNN-SMOTETomek	14.31%	40.07%	20.32%

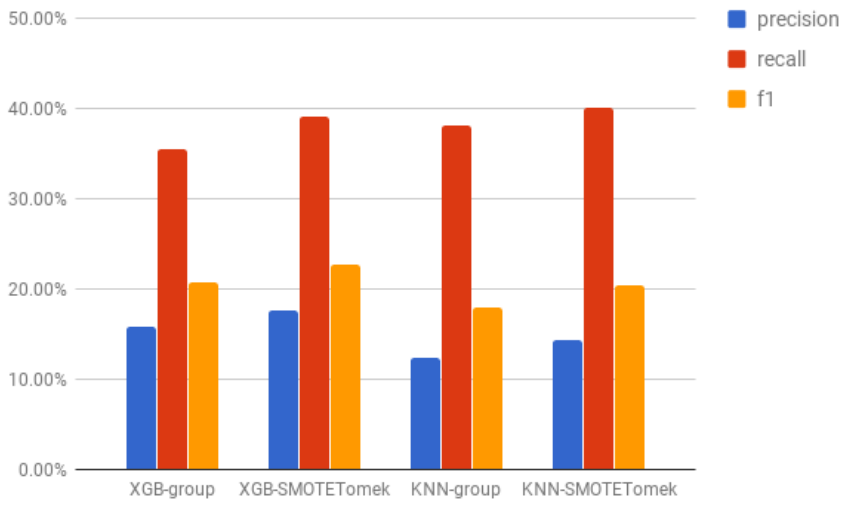


Figure 32: Classifiers tested on TED Talk subtitle Dataset in Real Use Case

7.3.4 Experiment 3.3: System Performance Evaluation

Finally, this experiment is going to evaluate the performance of the two additional component, namely sub-sentence generating component and topic relatedness filter component, as well as the complete TED Talk claim detection system. The result of this experiment, along with the result of experiment 2.3, can answer the fourth question mentioned before.

Experiment Setup

Figure 33 shows the procedure of this step. The evaluation process also simulates the real use case of the system. Each subtitle along with its topic was input into the system. The system will extract sentences containing context dependent claims from the subtitles. Precision and recall will be calculated each subtitle, and the overall precision and recall are the average of value measured among all 10 subtitles.

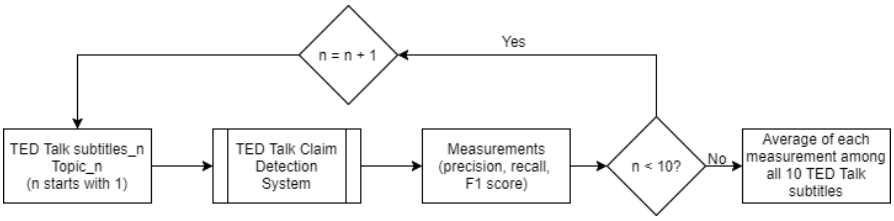


Figure 33: A procedure of this experiment

Some of the classifiers, built on the Wikipedia dataset in experiment 1, have been proved that can not be used in the system. According to experiment 1 which proved that the data balancing is needed, classifiers built on imbalanced data will not be used in TED Talk claim detection system. Meanwhile, experiment 2 shows that the only sentence component features classifier family might be able to have a relatively high precision on TED Talks subtitles while other classifiers will fail. Thus, in this experiment, only the performance of systems with the sentence component feature classifiers are checked.

Expected Outcome

If these components are working properly, systems with both components should have better results than those with only one of the component. Systems without any of the component should perform the worst. Also, the complete system should be able to extract sentences containing context dependent claims from a given TED Talk subtitle with high precision and recall.

Result of the Experiment

Table 14 and Figure 34 shows the result of applying the sub-sentence generating component only. Topic relatedness filter component will not be applied in this experiment. By splitting a complex sentence into sub-sentences, this component can help classifiers detect more target sentences from the given TED Talk subtitle. The recall of classifiers raises from 40% to more than 80%. During the experiment, the classifiers can only detect simple sentences such as "Gender equality is good for countries.". Longer sentences such as "So, what we found is something really important, that gender equality is in the interest of countries, of companies, and of men, and their children and their partners, that gender equality is not a zero-sum game." cannot be extracted. Meanwhile, when using the sub-sentence generating component, this sentence, which is labeled as a claim by annotators, will be included in the output of the classifier. This result is similar the the result in experiment 2.3. In other words, this component can help in increasing the recall of a classifier that are applied to both TED Talk subtitles and persuasive essays. Thus, we can assume that this component can indeed help the classifier to detect more sentences containing context dependent claims.

Table 14: Classifiers with sub-sentence generating component

	precision	recall	f1
XGB-group	10.91%	89.15%	19.38%
XGB-SMOTETomek	15.92%	85.72%	26.85%
KNN-group	10.57%	94.99%	18.72%
KNN-SMOTETomek	12.11%	93.45%	20.98%

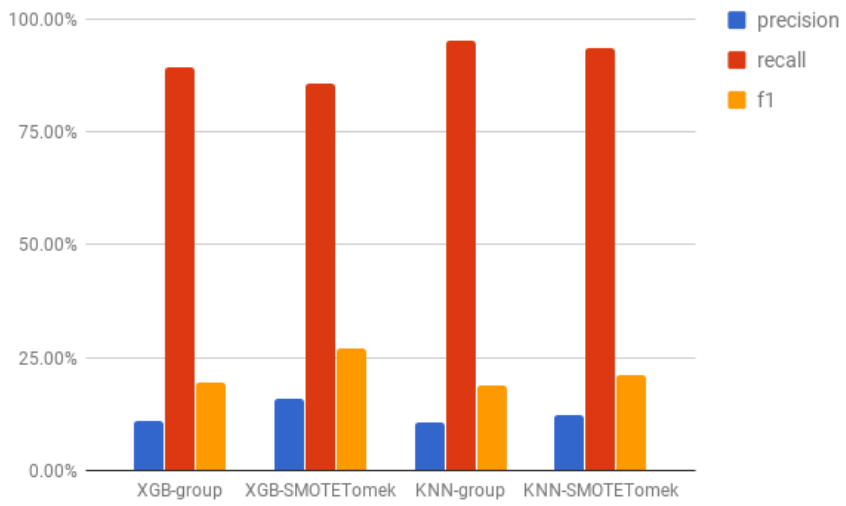


Figure 34: Classifiers with sub-sentence generating component

Table 15 and Figure 35 shows the performance of classifiers when only topic relatedness filter component are applied. The sub-sentence generating component will not be used. In this experiment, we checked both cascade strategy and linear combination strategies. Both of them can improve the precision of the classifier significantly. According to the result, a classifier with xgboost classification algorithm and SMOTETomek data balancing algorithm now has the best precision. Also, this result is similar to the result of experiment 2.3. This component can improve the precision of classifier on both the TED Talk subtitle dataset and the persuasive essays dataset. Therefore, we can assume that this component can indeed help classifiers extract sentences containing context dependent claims from a given TED Talk subtitle more precisely. Thus, to improve the performance of the classifiers, sub-sentence generating component and topic relatedness filter component are necessary.

Table 15: Result of Classifiers using Topic Relatedness Filter Component

		precision	recall	f1
Cascade	XGB-group	40.86%	30.62%	30.74%
	XGB-SMOTETomek	50.17%	13.11%	20.01%
	KNN-group	12.63%	20.27%	15.02%
	KNN-SMOTETomek	13.62%	18.31%	14.96%
Linear Combination	XGB-group	20.41%	39.97%	24.72%
	XGB-SMOTETomek	37.38%	20.63%	22.95%
	KNN-group	28.87%	51.37%	28.13%
	KNN-SMOTETomek	23.51%	51.71%	30.06%

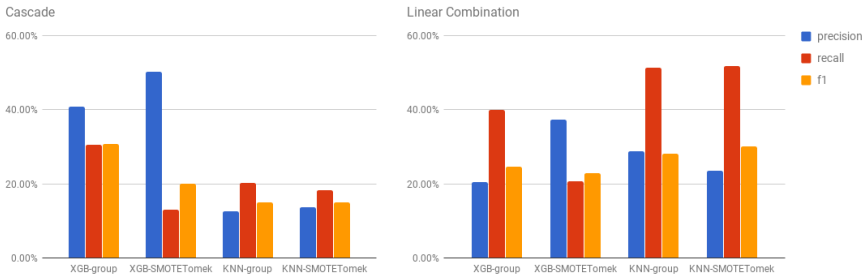


Figure 35: Result of Classifiers using Topic Relatedness Filter Component

Table 16 and figure36 shows the performance of the system on TED Talk subtitles dataset. Four different classifiers are checked. Both sub-sentence generating component and topic relatedness filter component are applied. We also checked the cascade strategy and linear combination strategy to combine two scores. According to the result, when using both additional components, the system now has much better precision and recall than using the classifier solely. Using cascade strategy can achieve better precision than using linear combination strategy. They also have higher F1 scores. When using linear combination strategy, we found it is hard to find a proper threshold for the system. The average relatedness between sentences and given topic, as well as the average probability score among all sentences, can differ a lot. This problem becomes more significant since we are using a small dataset. Each topic only has one TED Talk related to it. For example, when given the topic "germs", the highest relatedness score between topic and sentences in the relevant TED Talk is only 0.62. However, when given the topic "gender equality", the highest relatedness score is 0.79. In the TED Talk that is talking about "gender equality", the term "gender equality" is directly mentioned in the sentences such as "Gender equality is good for countries.". Also, this sentence receives a higher probability score since it follows the most basic structure of a claim. Meanwhile, the relatedness score between sentence "The microbes on your skin can help boost your immune system." and the topic "germs" is only 0.62, which is the highest relatedness score among all sentences in the TED Talk. The term "germs" never appears in the whole TED Talk subtitles. Instead, the synonyms of "germs" such as "bacteria", "viruses" and "fungi" are used.

Table 16: Performance of TED Talk claim detection system with different classifiers

		precision	recall	f1
Cascade	XGB-group	45.28%	55.38%	44.11%
	XGB-SMOTETomek	51.36%	41.81%	42.05%
	KNN-group	47.48%	48.09%	41.72 %
	KNN-SMOTETomek	53.17%	48.26%	45.77%
linear combination	XGB-group	20.17%	55.38%	29.57%
	XGB-SMOTETomek	20.72%	87.97%	30.34%
	KNN-group	34.36 %	39.88%	33.94%
	KNN-SMOTETomek	22.12%	88.48%	32.88%

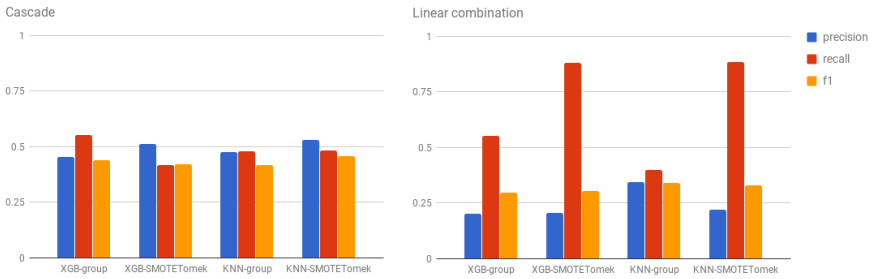


Figure 36: Result of the system using 2 different combine strategy

However, during this experiment, we found that when using the linear combination strategy, a sentence with a higher score is more likely to be a sentence containing a context dependent claim. Thus, we rank the sentences with their combined scores. By taking the top 5 sentences as the output of the system, we managed to get a much higher precision and f1 score. We take top 5 sentences since we found that the f1 score of the system reaches the top when taking the top 5 sentences. Table 17 and figure 37 shows the result of the performance of the system using this "TOP-5" result selection strategy. This strategy doesn't need predefined threshold and can improve the precision of the system using linear combination strategy. Compared with result in the table 16. The precision increased around 20%.

Table 17: Performance of system with linear combination strategy, use only top 5 sentences

	precision	recall	f1
XGB-group	50.00%	32.01%	35.91%
XGB-SMOTETomek	56.00%	34.31%	40.08%
KNN-group	54.00%	33.07%	38.42%
KNN-SMOTETomek	56.00%	33.65%	39.45%

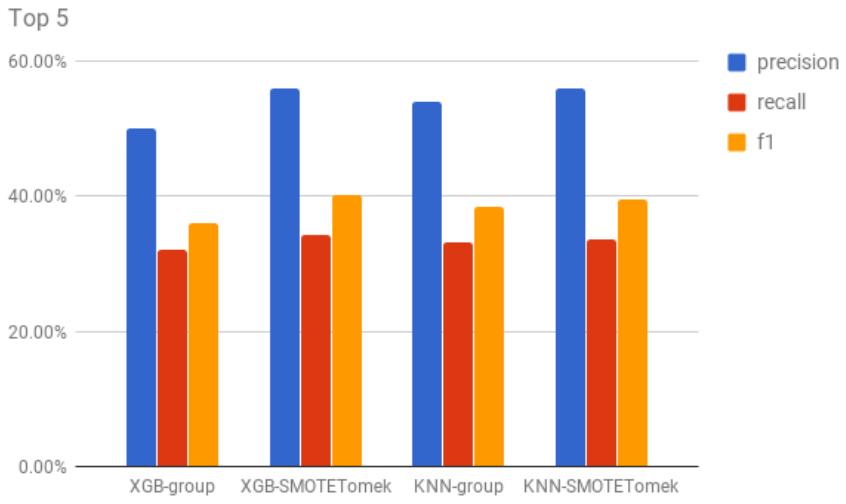


Figure 37: Result of the system using linear combination strategy, only top 5 sentence are taken

7.3.5 Conclusion of Experiment 3

According to these three experiments described in this section, we first proved that the result of annotating TED Talk subtitle with crowdsourcing techniques is acceptable but not highly

reliable. According to our experiment, the approximate average recall of the result given by each annotator is only around 50%. The result may not be able to include all sentences containing context dependent claims from the given texts. But, the average precision is 80%. by selecting sentences that are extracted by at least 2 annotators, the result is highly accurate. Nearly no sentence which doesn't contain context dependent claims are miss-labeled. When testing the performance of our system on this dataset, we can prove that sentences extracted by the system are indeed sentences containing context dependent claims. Also, the average agreement score among all annotator is 0.35, which is only slightly lower than 0.39, which is the baseline published by IBM. This score proved that annotators have a common understanding of the task. In other words, annotators are doing the same job and looking for the same kind of sentences. The design of this annotation task is successful.

Next, we proved that the performance of our claim detection system is acceptable. By applying sub-sentence generating component and topic relatedness filter component, we managed to achieve a precision higher than 50%. Although the precision may not be high enough, the random baseline is only 10%. Also, we found that some sentences that are extracted by our system, which should be considered as sentences containing context dependent claims, are not included in the result get from crowdsourcing annotation. For example, the sentence "Cyber weapons do not replace conventional or nuclear weapons – they just add a new layer to the existing system of terror." is considered as a sentence containing a context dependent claim by our system but is not extracted by annotators. As we introduced before, the dataset may not include all sentences containing context dependent claims in all 10 TED Talk subtitles. The precision of the system we measured should be lower than it's real performance. Also, the classifiers used in this system are trained on the Wikipedia dataset. Since the writing style of TED Talk subtitles differs a lot from Wikipedia articles. We believe that the performance of the system can be further improved by training the classifier on a large, well-annotated TED Talk subtitle dataset.

7.4 Experiment 4: Claims as "Teasing Texts"

In this experiment, we want to check if the sentences containing context dependent claims can improve the performance of content-based recommendation system. However, due to the fact that we only have 10 annotated TED Talks and no user data. Building a proper recommendation system would be impossible at this stage. Annotating more TED Talks and gathering user data will require huge efforts and can't be done in a short time. Thus, this experiment will only check if the claims extracted by our system can indeed be a better "teasing texts" compared with the manually written description of the TED Talk as well as the short summaries of the TED Talks.

Assuming that a user is interested in a certain topic and the recommendation system recommended an interesting TED Talk to the user. To convince the user that this is indeed a good TED Talk that worth spending some time watching it, three different types of "teasing texts" are used. The most common and easiest way to do this is to use the description of the TED Talk directly. Also, another popular solution is to use text summarization. Text summarization will generate a shorter version of the subtitle of the given TED Talk which contains the main idea of the TED Talk. This summarization could be a good choice. And finally, this thesis introduces another approach that is using the sentences containing context dependent claims extracted from the TED Talk subtitle. Since the claim, especially the main claim, holds the most important idea that this TED Talk wants to share, it can be extremely persuasive that can convince users to watch the recommended TED Talk. This experiment is going to answer the final question mentioned before.

Also, during this experiment, we found that using the claims extracted by our system can improve the result of keywords searching. The result will be more precise compared with the result of searching on the full subtitles.

7.4.1 Experiment 4.1: User Experiment About Using Claims As "Teasing Texts"

We mentioned that sentences containing context dependent claims are more powerful in convincing users to accept the recommendation results. This experiment is designed to check if

the sentences containing context dependent claims are indeed more persuasive and more user-friendly compared with other texts. In this experiment, we compare the sentences containing context dependent claims with TED Talks' descriptions as well as the short summaries.

Experiment Setup

In this experiment, we get the descriptions from TED.com. Each TED Talk used in this experiment has a manually written Text summarization.

Summaries are generated by the cutting-edge text summarization technique since it's the most commonly-used method for this task. The text summarization will be done in Python with a Python package called Gensim. When doing the summarization, we need to specify the average length of summaries we want. We use the average length of the descriptions, which is 68 words.

The claim detection system uses a sentence component feature classifier with the KNN classification algorithm and SMOTE+Tomek data balancing strategy. This system got the best result in experiment 3. For each TED Talk, only sentence with the highest score are used. This sentence is the only that are most likely to contain context dependent claim and most relevant to the given topic considered by our claim detection system. Experiment 3 indicates that the precision of the system is only 50%, using more sentences will increase the probability of including a sentence without any claim. Using multiple claims will also decrease the continuity of the whole text. Since claims are located separately in a subtitle, simply putting them together without any modification may make the whole texts less readable. Also, we think that showing only one claim to users is enough to convince them and the "teasing text" shouldn't be too long.

We created a questionnaire form which contains 10 questions. Each question checks one of the ten annotated TED Talk subtitles. Participators are asked to give scores to three different types of "teasing texts" that are the description, the text generated by text summarization technique and the sentence containing context dependent claim extracted by our system. Figure 38 shows an example of one question in our questionnaire form. In each question, we assume participator is interested in a certain topic and our system has found a TED Talk that is relevant to that topic. Then, we present three different types of "teasing texts" to the participator. The participator will read these texts and give scores from 1 to 10. A score represents how persuasive this text is to the participator as well as how friendly the user experience is. In other words, participators should consider whether they will be convinced that this is indeed an interesting and attractive TED Talk by reading the given text. A higher score means the text is more likely to persuade the participator to watch the recommended TED Talk. For each TED Talk, the score to a "teasing text" is the average of the scores given by all annotators. And the score to one type of "teasing texts" is the average score among all texts that belong to this type.

Expected Outcome

As we introduced before, a higher score means that this type of "teasing texts" is more persuasive and user-friendly. Thus, if our approach is indeed better than using the descriptions or the texts generated by text summarization, our approach should get a higher score. Also, in order to prove that these "teasing texts" can convince users to watch the recommended TED Talk, it is necessary to set a baseline. The score to a type of "teasing texts" should be higher than the baseline. However, there isn't any baseline set in previous researches, we use the score of using the descriptions directly as the baseline since this is the most straightforward and easy-to-get texts. Every TED Talk has its own manually generated description. If another type of "teasing texts" has higher score, it will indeed provide additional value that can improve the performance of the recommendation system.

Result of the Experiment

We got 15 responses in total. Table 18 and figure 39 show the distribution of scores of each type of "teasing texts" among all 10 TED Talks. The average score of using the manually generated descriptions directly is 6.88. The score of using the sentences containing context dependent claims

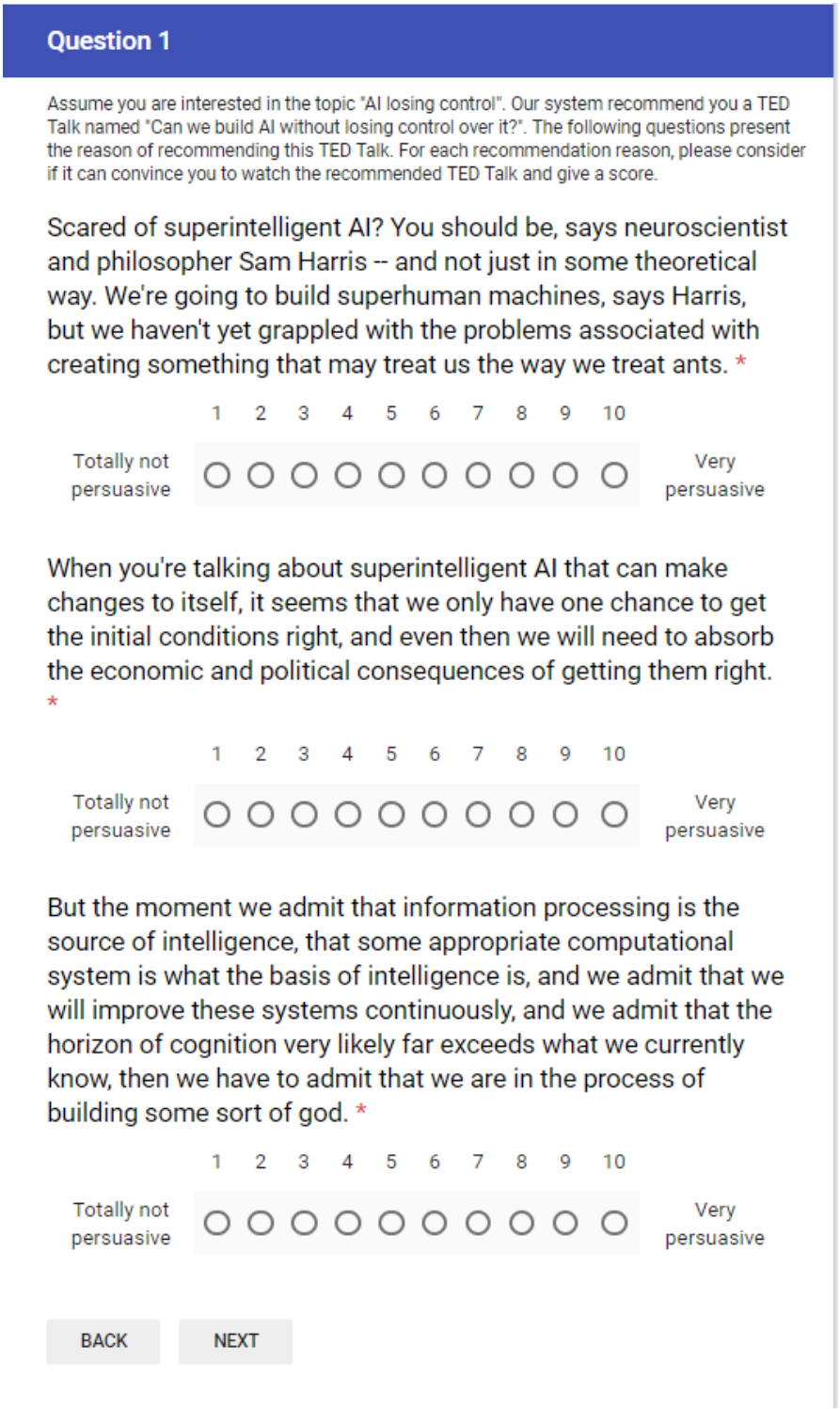


Figure 38: An example of question used in our survey

is 6.47. This score is close the score of using the manually generated descriptions. And the score of using the text summarization is only 5.54 which is significantly worse than other 2 types. Also, text summarization has the highest standard deviation, which means the performance of using text summarization can be unstable. Sometimes it can perform even worse. Descriptions has the lowest deviation since they are manually generated with requires human efforts. Our approach, that is using the claims extracted by our system, can perform similar with using the descriptions. Also, the performance of our approach is better than performance of using text summarization techniques.

Table 18: Distribution of scores of each type of "teasing texts" among 10 TED Talks

	Mean	STD
Description	6.88	0.30
Claim Detection	6.47	0.36
Text Summarization	5.54	0.40

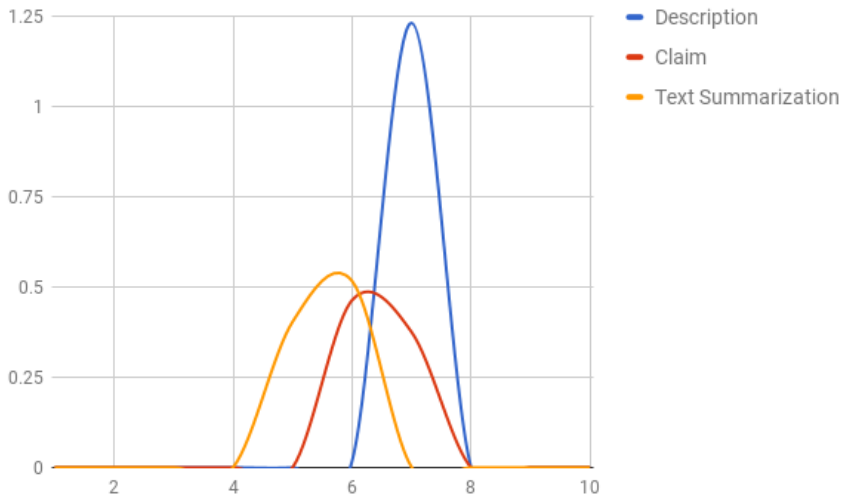


Figure 39: Distribution of scores of each type of "teasing texts" among 10 TED Talks

The sentences containing context dependent claims can be much shorter than the descriptions and summaries. For example, the description of TED Talk "How cyberattacks threaten real-world peace" is "Nations can now attack other nations with cyber weapons: silent strikes on another country's computer systems, power grids, dams that leave no trace behind. (Think of the Stuxnet worm.) Guy-Philippe Goldstein shows how cyberattacks can leap between the digital and physical worlds to prompt armed conflict – and how we might avert this global security hazard.". And the sentence with the context-dependent claim we extracted is "So military technologies can influence the course of the world, can make or break world peace – and there lies the issue with cyber weapons.". Table 19 and figure 40 shows the average length of the descriptions, claims and summaries, as well as the standard deviation. The average length of claims are only 23 words. And the shortest claim contains only 8 words. It could make the claims more user-friendly since users will spend less time on reading these texts.

Table 19: Distribution of the length of sentences in each type of "teasing texts"

	Mean	STD
Description	68	11.49
Claim	23	10.52
Text Summarization	64	10.71

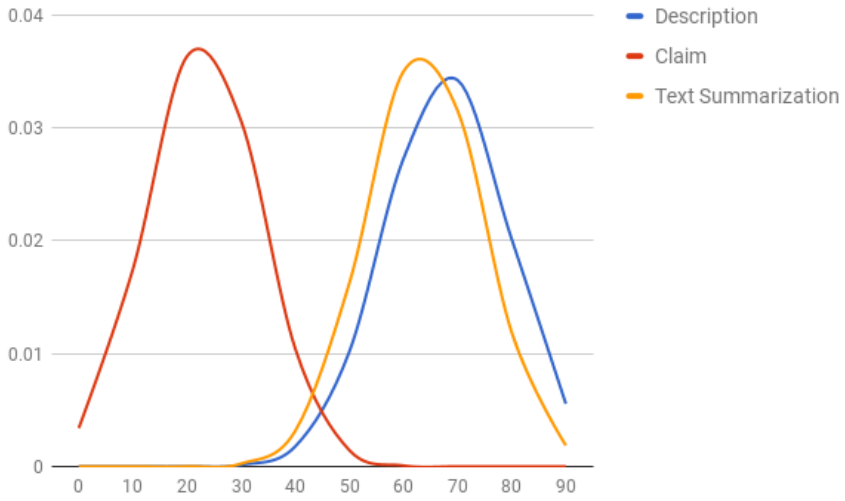


Figure 40: Distribution of the length of sentences in each type of "teasing texts"

During this experiment, we found that there are several things that could lead to a better performance once improved. The sentence we used as the "teasing text" is the one with the highest score among all sentences extracted by our system, which means this sentence is not only most likely to contain a claim but also the most relevant to the topic. However, we can't prove that the sentence we chose contains the main claim. The main claim, usually, is the most important idea this TED Talk wants to share. It makes the main claim the most suitable sentence to be used as the "teasing text". Thus, we believe that by improving the system that can extract the main claim of the TED Talk, using sentence containing context dependent claim can further improve the performance of the recommendation system.

7.4.2 Experiment 4.2: Performance of Claims Based Keywords Searching

During this experiment we found that using the sentences containing context dependent claims can improve the result of keyword searching. This section will explain the setup of this experiment and analyze the result.

Experiment Setup

We built 2 keywords searching systems using elasticsearch. The 10 annotated TED Talks are indexed. Two systems use the same setting and features. The only difference between 2 systems is the documents used. The first system indexes the full subtitles while the second one only indexes the sentences containing context dependent claims extracted by our claim detecting system. The claim detection system uses a classifier with KNN classification algorithm and SMOTE+Tomek data balancing strategy since it has the best performance in experiment 3. As we mentioned before, TED Talks we annotated are selected under 10 different topics. These topics are used as the query in this experiment. The output of a keywords searching system is relevant TED Talks and their scores. The scores indicate the confidence of returning this result.

Table 20: Some Results of both recommendation systems

Topic	Claim based		keywords search	
	TED Talk	Score	TED Talk	Score
Ai losing control	Can we build AI without losing control over it?	1.79	Can we build AI without losing control over it?	2.28
			Big data is better data	0.87
big data	Big data is better data	4.69	Big data is better data	2.99
	Climate change is happening. Here's how we adapt	1.03	How YouTube thinks about copyright	2.09
	What fear can teach us	0.70	Climate change is happening. Here's how we adapt	1.51
	Why gender equality is good for everyone	0.27	Why gender equality is good for everyone	1.20
			The future of early cancer detection?	0.95
			We're covered in germs. Let's design for that.	0.84
			What fear can teach us	0.58
cancer detection	The future of early cancer detection?	4.61	The future of early cancer detection?	3.50
	Big data is better data	0.53	Big data is better data	1.24
			Can we build AI without losing control over it?	0.59

Expected Outcome

When given a topic, a better system should have a more accurate result. It should not only return all the TED Talks that are highly relevant to the given topic but also omits other irrelevant TED Talks. By comparing the results returned by both system, we can check if we can improve the keywords searching result by using only the claims extracted by our claim detection system.

Result of the Experiment

According to the experiment. Both keywords searching systems can return the most-wanted TED Talk. When given a topic, the TED Talk which is the most relevant to it will always get the highest score. However, the system that uses sentences containing context dependent claims can omit more irrelevant TED Talks. For example. When given to topic "cancer detection", the claim based system will return two TED Talks, including "The future of early cancer detection?" and "Big data is better data.". The first TED Talk got a score of 4.61 while the second one only got 0.53. The second TED Talk can be ignored due to its extremely low score. The result of the system that uses full subtitles contains one more TED Talk called "Can we build AI without losing control over it?" which is not relevant to the topic "cancer detection". Meanwhile, the score of "The future of early cancer detection?" reduced to 3.50 and the score of "Big data is better data" increased to 1.24. Table 20 shows some results of 2 keywords searching systems. According to this result, using only the claims leads to a more precise searching result. Because when using the sentence containing context dependent claims, only main ideas are considered by the system. However, this experiment only checks the performance on 10 TED Talk subtitles. The result may contain bias which makes the result less reliable. If we want to draw a more solid conclusion, more data is needed.

7.4.3 Conclusion of Experiment 4

In conclusion, the claim sentences we extracted from the subtitles are more persuasive compared with the summaries generated by text summarization techniques. Although the they perform slightly worse than using the descriptions, the claims are automatically extracted from the

subtitles and can save a lot of human effort. Using these claims as the "teasing text" is feasible and can motivate students to watch the recommended TED Talks. Also, we found that extracting claims from a given texts can help getting a better result in information retrieval. When using the claims only, keywords searching can produce more precise result.

7.5 Threats to Validity

During the experiment, we found that the following issues may have bad influence on the validity of the whole experiment.

7.5.1 Factors Which Jeopardize Internal Validity

The TED Talk subtitle dataset built in this experiment isn't highly reliable. We use only 10 TED Talks which mean the size of this dataset is rather small. Also, the TED Talk subtitles are randomly selected. When the sample size is small, randomization may lead to Simpson's paradox. Which means the TED Talk subtitles we used in this experiment may not be able to represent the general situation. The chance that the system only works on the selecting TED Talk subtitles cannot be ignored. We tried to solve this problem by using both the persuasive essays dataset and the TED Talk subtitle dataset to validate the performance of the system. There are some differences between persuasive essays and TED Talk subtitles. For example, the claims in persuasive essays are context independent claims. Also, unlike the TED Talk subtitles which are the direct record of the speech, persuasive essays are written in formal writing English. These differences make the result less reliable. In experiment 4, we only managed to get 15 responses and again only checked 10 TED Talks. The result of this experiment may suffer the same problem. To make the result more reliable, more samples are needed.

Also, the performance of the annotator is acceptable but not highly reliable. When relying on crowdsourcing, the annotators often lack the expert knowledge which makes the result worse than those using expert annotators. In this experiment, although the average agreement is close to the baseline published by IBM who uses professional annotators, the estimated recall of the result is only 50%. Which means the annotators may miss several sentences containing context dependent claims. When using this dataset as the testing set, the reliability of evaluation is reduced since some target sentences extracted by the system are considered as sentences without context dependent claims by annotators.

7.5.2 Factors Which Jeopardize External Validity

In this experiment, the thresholds used in the topic relatedness filter component are set based on existing dataset. For example, when using the cascade strategy, the threshold of the relatedness score is based on the average of relatedness scores of sentences in the persuasive essays dataset and the Wikipedia dataset. However, we cannot prove that these thresholds are the most suitable threshold when applying the system to TED Talk subtitles. Thus, the measurements of the performance of the system using these thresholds may not represent the true performance of the system when applying to TED Talk subtitles. We might be able to get a better result by using different threshold. Also, because the TED Talk subtitle dataset used in this experiment is quite small, optimize the thresholds with the testing results on this dataset may lead to higher bias. Thus, this experiment only proved that the thresholds we set can lead to a relatively good performance of the system. We cannot prove that the thresholds used in this experiment are the best thresholds for the system.

8 Conclusion and Future Work

8.1 Conclusion

This thesis implemented several claim classifiers based on three different categories of approaches. All classifiers are trained on the largest open-source dataset built on Wikipedia articles. We first

check the performance of the classifiers on the Wikipedia dataset to evaluate if our implementation is successful. By using various types of classification algorithms and data balancing algorithms, all classifiers have been successfully implemented.

Next, this thesis tested all classifiers on the persuasive essays dataset and the TED Talk subtitles dataset. This time, only sentence component feature classifiers still hold acceptable performances. It is the only category of classifiers that can overcome the cross-domain learning problem. Sequential pattern mining classifiers and tree-kernel classifiers are considered overfitted. They work extremely well on the Wikipedia dataset but fail at the persuasive essays dataset as well as the TED Talk subtitles dataset.

Then, this thesis provides a successful approach to extract claim from TED Talk subtitles with classifiers trained on the Wikipedia dataset. two additional components aiming at improving the precision and recall separately are added to the system. This system is able to detect sentence containing context dependent claims from a given TED Talks with acceptable precision and recall. The precision and recall are much higher than the random baseline.

Finally, we checked the performance of using sentences containing context dependent claims as the "teasing texts". Indicating sentences containing context dependent claims can be an alternative solution of maually written descriptions. And they perform better than text generated by text summarization technique.

8.2 Future Work

However, there are still some improvements need before using the system in practice. The most important will be speeding up the system. Now the system takes around 1 second to analysis one sentence. When dealing with a complex sentence which may have a lot of sub-sentence, the prediction process can take up to 5 seconds. A single TED Talk subtitle may contain over 200 sentences, which may take minutes before finishing the process. Meanwhile, there are over 2000 TED Talks published. Analyze all of them will take hours.

Also, researchers now start to solve natural language processing problems with deep learning techniques. It is worthwhile to try some deep learning approach, which might be able to improve the performance of the whole system.

Building a large dataset of TED Talk may also improve the performance. Since the classifiers are trained based on the Wikipedia dataset and the writing style is quite different between Wikipedia articles and TED Talk subtitles, the training set could limit the performance of this system. Therefore, building an extensive dataset by annotating all available TED Talk subtitles with expert knowledge could make the classifier more precise and able to extract more claim from the TED Talk subtitles. However, this could take years.

Finally, as we introduced in section 7.4. We can further improve the "teasing texts" by extracting the main claim from the TED Talk subtitle. This requires us to extract the relationships among all claims and premises we found which could be extremely hard. However, according to our experiment, using the sentence containing context dependent claim as the "teasing text" is indeed better than using the summaries an is only slightly worse than manually written descriptions. It's worth to put more effort on selecting a better claim.

After solving these problems, several research is needed to check whether using TED Talks can indeed motivate students. Since there isn't any research aimed to check the power of TED Talk in motivating students yet, the future work will be focused on evaluating the influence of TED Talks on motivating students.

References

[1] E. Aharoni, A. Polnarov, T. Lavee, D. Hershcovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *ArgMining@ ACL*, pages 64–68, 2014.

[2] J. T. Austin and J. B. Vancouver. Goal constructs in psychology: Structure, process, and content. *Psychological bulletin*, 120(3):338, 1996.

[3] E. Cabrio and S. Villata. Natural language arguments: A combined approach. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 205–210. IOS Press, 2012.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[5] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[6] T. Elhassan, M. Aljurf, F. Al-Mohanna, and M. Shoukri. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Journal of Informatics and Data Mining*, 2016.

[7] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[8] E. Florou, S. Konstantopoulos, A. Koukourikos, and P. Karampiperis. Argument extraction for supporting public policy formulation. In *LaTeCH@ ACL*, pages 49–54, 2013.

[9] M. E. Ford. *Motivating humans: Goals, emotions, and personal agency beliefs*. Sage, 1992.

[10] T. Goudas, C. Louizos, G. Petasis, and V. Karkaletsis. Argument extraction from news, blogs, and social media. In *SETN*, pages 287–299. Springer, 2014.

[11] N. Grandgenett. Ted: Ideas worth spreading. *Mathematics and Computer Education*, 46(1):76, 2012.

[12] I. Habernal, J. Eckle-Kohler, and I. Gurevych. Argumentation mining on the web from information seeking perspective. In *ArgNLP*, 2014.

[13] T. Heafner. Using technology to motivate students to learn social studies. *Contemporary Issues in Technology and Teacher Education*, 4(1):42–53, 2004.

[14] S. Hidi and J. M. Harackiewicz. Motivating the academically unmotivated: A critical issue for the 21st century. *Review of educational research*, 70(2):151–179, 2000.

[15] H. Houngbo and R. E. Mercer. An automated method to build a corpus of rhetorically-classified sentences in biomedical texts. In *ArgMining@ ACL*, pages 19–23, 2014.

[16] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim. Context dependent claim detection. 2014.

[17] R. Levy, L. Ein-Dor, S. Hummel, R. Rinott, and N. Slonim. Tr9856: A multi-word term relatedness benchmark. In *ACL (2)*, pages 419–424, 2015.

[18] P. Li, Q. Zhu, and G. Zhou. Argument inference from relevant event mentions in chinese argument extraction. In *ACL (1)*, pages 1477–1487, 2013.

[19] M. Lippi and P. Torroni. Context-independent claim detection for argument mining. In *IJCAI*, volume 15, pages 185–191, 2015.

[20] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):10, 2016.

[21] M. Lippi and P. Torroni. Margot: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303, 2016.

[22] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[23] F. Masseglia, M. Teisseire, and P. Poncelet. Sequential pattern mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1028–1032. IGI Global, 2005.

[24] M. C. McCord, J. W. Murdock, and B. K. Boguraev. Deep parsing in watson. *IBM Journal of Research and Development*, 56(3.4):3–1, 2012.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[27] R. Mochales and M.-F. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[28] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM, 2007.

[29] C. H. Mooney and J. F. Roddick. Sequential pattern mining–approaches and algorithms. *ACM Computing Surveys (CSUR)*, 45(2):19, 2013.

[30] A. Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*, volume 4212, pages 318–329. Springer, 2006.

[31] P. R. Pintrich. The role of goal orientation in self-regulated learning. *Handbook of self-regulation*, 451:451–502, 2000.

[32] R. Rinott, L. Dankin, C. A. Perez, M. M. Khapra, E. Aharoni, and N. Slonim. Show me your evidence-an automatic method for context dependent evidence detection. In *EMNLP*, pages 440–450, 2015.

[33] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis. Argument extraction from news. In *ArgMining@ HLT-NAACL*, pages 56–66, 2015.

[34] U. Schiefele. Interest, learning, and motivation. *Educational psychologist*, 26(3-4):299–323, 1991.

[35] J. M. Shaughnessy and T. M. Haladyna. Research on student attitude toward social studies. *Social Education*, 49(8):692–95, 1985.

[36] C. Stab and I. Gurevych. Annotating argument components and relations in persuasive essays. In *COLING*, pages 1501–1510, 2014.

[37] C. R. Stefanou, K. C. Perencevich, M. DiCintio, and J. C. Turner. Supporting autonomy in the classroom: Ways teachers encourage student decision making and ownership. *Educational Psychologist*, 39(2):97–110, 2004.

[38] Y. Sun, A. K. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[39] S. E. Toulmin. *The uses of argument*. Cambridge university press, 2003.

[40] K. C. Williams and C. C. Williams. Five key ingredients for improving student motivation. *Research in Higher Education Journal*, 12:1, 2011.

[41] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.