Fake News Detection: Automatic Verification of Facts

Master Thesis

Master in *Sciences and Technologies*,
Specialty in *Computer Science*,
Academic Course DECOL

Author
Jérémy BRESSAND

Supervisors
Konstantin TODOROV, LIRMM, CNRS,Univ. Montpellier
Mathieu ROCHE, TETIS, CIRAD
Stefan DIETZE, L3S Leibniz University Hanover

Master Thesis's Location
LIRMM UM5506 - CNRS, University of Montpellier

June 25, 2018

Abstract

Nous nous intéressons dans ce mémoire à la vérification automatique de la véracité d'un fait exprimé sous forme textuelle. Nous présentons tout d'abord les intérêts du domaine de recherche, les travaux connexes ainsi que la problématique. Puis nous expliquons notre approche et nos expérimentations. Nous terminons par présenter nos résultats et discuter des performances et limitations de nos travaux avant de conclure par un rappel sur le travail réalisé et une présentation des perspectives associées.

Abstract

In this work, we focus on the automatic fact-checking task over textual content. First, we introduce the research field interests, related work and our problematic. Then, we explain our approach and our experiment. Therefore, we present our results and discuss their potential and limitations. Finally, we conclude by summing up the work done and introduce future work.

Contents

1

Introduction

News appears in a continuous stream, increasing in volume and velocity with the growing influence of social media. While journalist has difficulties to quickly verify the veracity of this news, the research task of automatic fact-checking arouse interest.

One key challenge for researchers is to find data with experts annotations. Fact-checking websites contain news with associated veracity annotations given by journalists, therefore they meet researchers expectations.

But news is limited in information. To verify a piece of news, a journalist need to find other sources of information relevant with regards to this piece of news, hence researchers need to enrich news with relevant information.

In this work, we build a data set by crawling a fact-checking website and using a web search engine.

Furthermore, we introduce our approach for automatic fact-checking based on Text Mining and Machine Learning techniques. Our approach consists of a two-fold process, where first, irrelevant information is deleted using semantic representations and similarity measures.

In a second step, we apply machine learning techniques for binary classification on semantic representations of news and semantic representations of relevant information associated to news. We do not demonstrate superior performances for representations of relevant information associated to news over representations of news themselves.

The main contributions of this work are the data set enriched with web search engine results and the ground truth data set for the relevance discovery step.

The paper is structured as follows: Chapter 2 discusses related work with knowledge base approaches and unstructured data approaches. Chapter 3 provides the problem statement. Chapter 4 introduces what type of representations of our data we want to explore and an overview of our approach. Chapter 5 describes the experimental setup, followed by the presentation of results in Chapter 6 and we conclude in Chapter 7.

2

Related Work

In this chapter, we review related literature. We focus on two main groups of approaches for automatic fact-checking: *knowledge base* and *unstructured data* approaches.

2.1 Knowledge Base

Knowledge base approaches intend to collect and model structured knowledge. One possible approach to verify an information is to check if the information is present in a knowledge base, therefore these approaches can be useful for automatic fact-checking.

Some previous works collected knowledge from webmasters annotations on web pages [2, 1]. One of these works intend to augment a preexisting knowledge base [2] whereas the other intend to build a novel knowledge base [1].

A limitation of these works is the knowledge bottleneck, hence the coverage of knowledge is limited. These works focus on domain-restricted knowledge. For instance, [2] augment knowledge bases with information related to books, movies and persons.

Our approach is not concerned by this knowledge bottleneck issue if we consider that we can find information relevant with regards to any news using a query engine such as a web search engine.

2.2 Unstructured Data

Unstructured data approaches work on textual representation of news instead of structured representation and their goals are the same as the automatic fact-checking task [7, 5, 4].

But unstructured data approaches goals can be close to the automatic fact-checking task such as the check-worthiness task [3] which intend to detect information that deserve to be verified. For example, the sentence 'I ate an apple yesterday' does not deserve to be verified.

Another task is the relevant document discovery task [5, 6] which intend to associate to each news textual documents relevant with regards to this piece of news. For instance, the wikipedia page of Barack Obama containing the information that he was born in Honolulu is a relevant document with regards to the fact 'Obama was born in Kenya'.

Finally, the stance detection task [5, 6] intend to detect if a document support, refute, discuss or is not related to another document. If we look at the example that illustrates the relevant document discovery task, we observe that the wikipedia page refute the fact that Obama was born in Kenya.

Some previous works use web search engines to find information relevant w.r.t. news [5, 6].

One limitation of these works is that they need the entire content of documents but during our experimentation, we observe that many results from the web search engine are irrelevant and fetching web content associated to these results can be time and space consuming.

Our approach doesn't suffer this limitation because we don't need the content of documents to do relevance discovery, but only a small textual representation of the document's content provided by the web search engines.

3

Problem Definition

For the purpose of this work, we need to define several key concepts: facts, snippets, documents, relevant snippets and relevant documents.

Definition 1. *Fact:*
A fact is a text containing information that deserved to be verified. The date associated to a fact is the date when the fact was fact-checked on a fact-checking website. The label associated to a fact is an annotation about the veracity of the fact made by a editor from the fact-checking website.

A fact contains not enough information in itself for our automatic fact-checking models to assess its veracity. In order to find useful information associated to facts, we retrieve snippets using a web search engine.

Definition 2. *Snippet:*
A snippet is a small text describing a result from a web search engine query. The url and the title of a snippet are the url and title associated to the snippet by the web search engine. The snippet was extracted from the web document corresponding to the url.

Snippets are only small textual representations of documents which contain useful information related to facts.

Definition 3. *Document:*
A document is a text associated to a result from a web search engine query. The document is made by subtracting all text content from the web page from the url of the corresponding snippet.

The problem of using a web search engine is that many results are irrelevant, hence we need to identify relevant snippets.

Definition 4. *Relevant snippet:*
A relevant snippet with regards to a fact is a snippet with a similarity score with regards to this fact above a threshold fixed during our experiment.

Snippets are small textual representations of documents. Similarly, relevant snippets are small textual representations of relevant documents.

Definition 5. *Relevant document:*
A relevant document with regards to a fact is a document with an associated snippet relevant with regards to this fact.

Now that we have defined key concepts, we can define the task of automatic fact-checking:

Definition 6. *Automatic fact-checking:*
The automatic fact-checking task aims at predicting for each fact the corresponding label based on the fact or relevant documents with regards to this fact.

Our hypothesis during this work is that relevant documents contain useful information to predict the label of a fact. This work intend to demonstrate this hypothesis.

4

Features and Approach

First, we describe what type of representations or features associated to facts, snippets and documents we use in our approach. Then, we detail the overall approach of our work.

4.1 Features

To apply machine learning techniques on texts, we first need a structured representation of these texts. We refer to such structured representations as features.

We apply standard text-mining techniques to extract features from text, namely bag of word representation, TF-IDF representation and features associated with topic modelling techniques.

First, we will use the simplest text representation called bag of words.

Definition 7. *Bag of word representation (BOW):*
A standard representation of a text which counts the occurrence number of each word from a dictionary in the text. This representation does not take into account the order of words in a text.

Another text representation is the term frequency - inverse document frequency. This representation take into account frequencies associated to each words across individual texts and collection of texts.

Definition 8. *Term frequency - inverse document frequency representation (TF-IDF):*
A standard representation of a text intended to reflect the frequency of a word in the text in balance with the inverse frequency of this word in a collection of texts for each word in a dictionary. As the BOW representation, it does not take into account the order between words in a text.

All the above standard features do not capture the semantic of words. Topic modelling can build clusters of words sharing the same semantic by identifying words appearing frequently together in a collection of texts.

Definition 9. *Topic modelling techniques:*
Topic modelling techniques aim to discover hidden topics in a collection of texts and

Table 4.1: List of features.

Feature name	Category of feature
Bag of Words (BOW)	Standard text-mining representation
Term frequency - inverse document frequency (TF-IDF)	Standard text-mining representation
Latent Semantic Indexing (LSI)	Topic modelling representation
Latent Dirichlet Allocation (LDA)	Topic modelling representation
Random Projection (RP)	Topic modelling representation

the importance of each topic for each text of this collection, hence they aim to build a semantic representation of texts. We refer to topics as clusters of similar words.

Table 4.1 list all the features and their associated categories. All these features need a prior collection of words and identification numbers associated, namely a dictionary. The quality of these types of features depends on the quality of the dictionary. Therefore, we'll try different configurations of dictionaries by varying the number of words parameter. For each type of features, we will build different features based on different dictionaries.

Furthermore, the quality of topic modelling features depends on several parameters. First, we can set the number of hidden topics to discover as a parameter for topic modelling techniques. The choice of which topic modelling technique to use (e.g. Latent Semantic Indexing, Latent Dirichlet Allocation, etc.) may also have a significant impact on topic modelling quality.

4.2 Approach

Our approach for addressing the automatic fact-checking problem defined in Chapter 3 consists of two steps, namely relevant snippet discovery and binary classification as we see in figure 4.1. We describe these steps below.

Relevant snippet discovery

For each fact, we retrieve a set of snippets by querying a web search engine.

For each type of features and associated configurations, we extract a feature from each fact and each snippet.

Then, for each feature, for each snippet, we apply a similarity measure between the feature extracted from the snippet and the feature extracted from the associated fact, the corresponding similarity score will be associated to the snippet.

We focus on the similarity scores associated to the feature which give the best performances w.r.t. a metric and ground truth data.

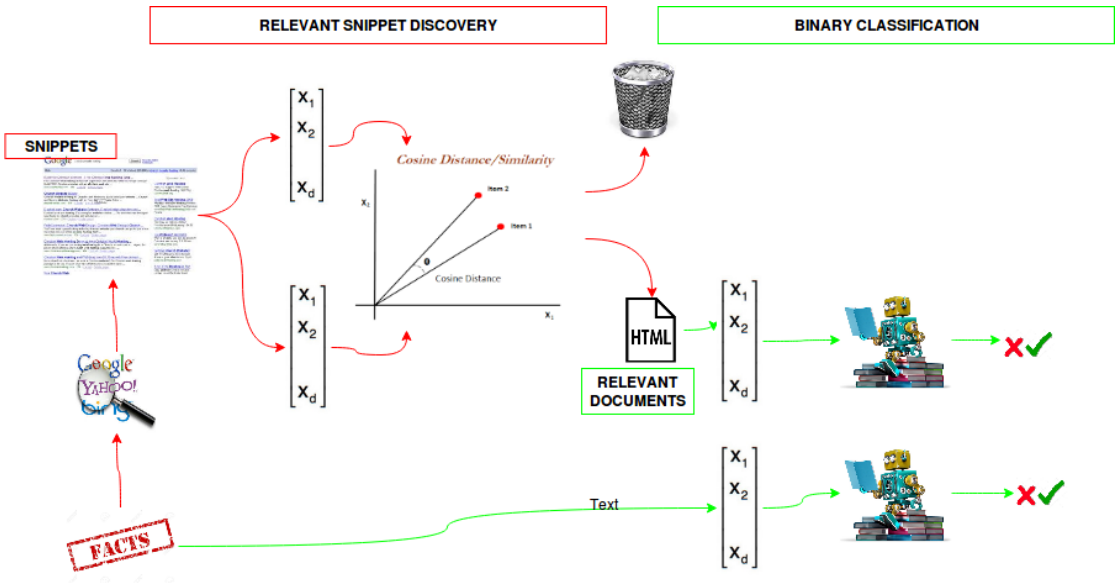

Figure 4.1: Twofold approach for automatic fact-checking.

We keep only snippets with similarity scores associated to the best feature for relevant snippet discovery above a threshold. We choose this threshold such as precision over ground truth data and number of relevant snippets kept are maximised.

Once we have filtered relevant snippets, we fetch the associated relevant documents by fetching web pages using urls from these snippets and extracting text content.

Binary classification

First, for each fact, we aggregate the associated top-k relevant documents ranked by their similarity scores in descending order into one single document by concatenating them.

If a fact has less than k documents then we delete this fact from the data. This k number of relevant documents is arbitrary and is a trade-off between taking into account a large number of documents per fact and keeping the maximum number of facts.

The idea behind aggregating documents is to homogenise our data, hence our machine learning techniques do not discriminate between facts with few documents and facts with many documents.

Once our data is homogenised, we start extracting features. Our goal is to compare performances between features extracted from facts and features extracted from documents. For each type of features and associated configurations, we extract the features for each fact and for each relevant aggregated documents.

Finally, we train and evaluate several binary classifiers on each feature. The maximum performance among classifiers is associated to the feature.

Our goal is to demonstrate that features extracted from relevant aggregated documents have better performances than features extracted from facts.

5

Experimental Setup

First, we detail how we collect our data and what transformations we apply on this data. Then, we describe how we build a ground truth data set for the relevant snippet discovery step of our approach and what metrics we use for relevant snippet discovery and binary classification steps. Finally, we introduce the configurations and baselines of our approach.

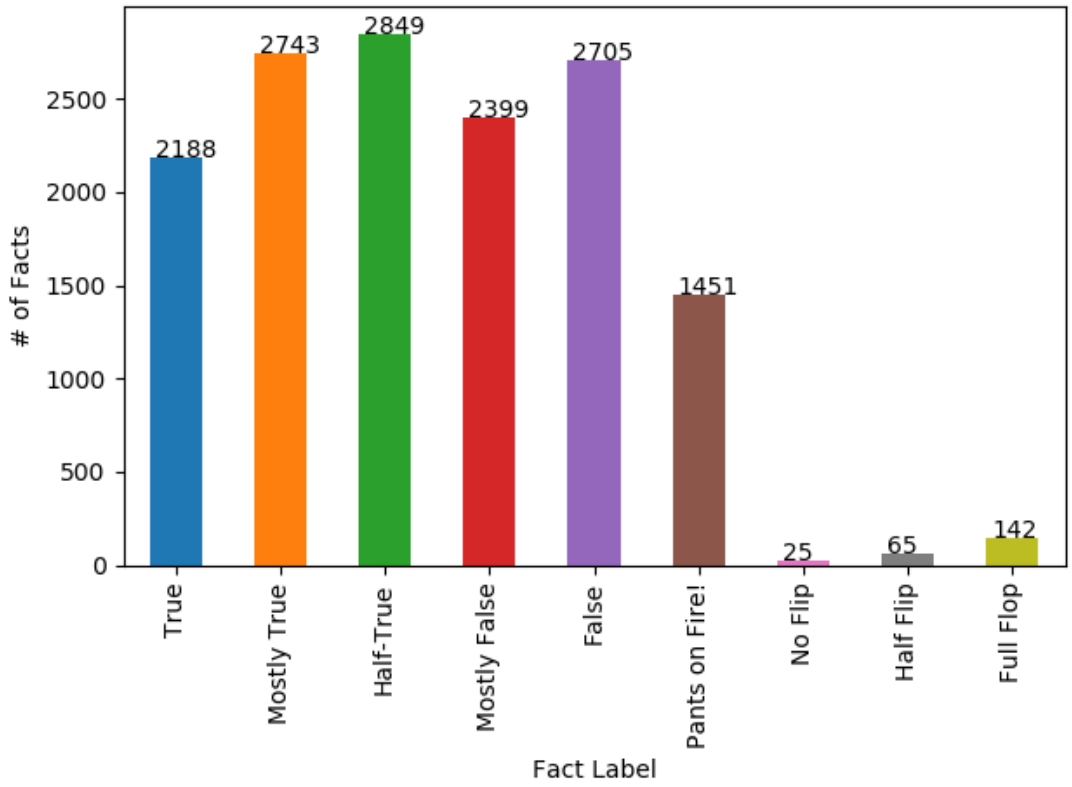

Figure 5.1: Number of facts per label associated.

5.1 Data

Our data combines facts, snippets and documents. For each compound of our data, we describe below how we collect the associated data and how we transform it.

Data acquisition

Facts: We fetch facts from the website PolitiFact[1]. We can see on figure A.1 a screenshot of the website and information associated to facts.

PolitiFact is run by journalists and owned by a non-profit organization[2].

On figure 5.1 we can see how many facts we have per label. 'No Flip', 'Half Flip' and 'Full Flop' are labels describing to what extent the author of the fact has changed his discourse whereas the other labels assess the veracity of the fact itself. The meaning of these labels is more detailed on their website[3]. We focus only on true and false claims to simplify our work.

[1] http://www.politifact.com/truth-o-meter/statements/

[2] http://www.politifact.com/truth-o-meter/staff/

[3] http://www.politifact.com/truth-o-meter/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/

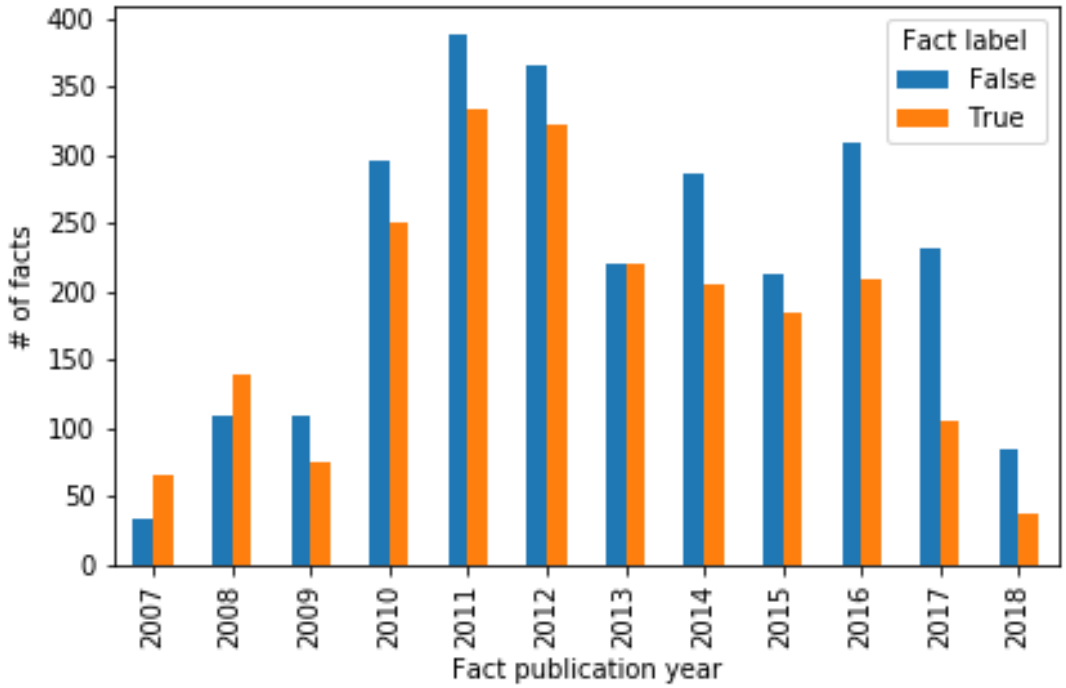

Figure 5.2: Distribution of true and false facts from 2007 to 2018.

On figure 5.2 we can see the number of true and false facts per year from the beginning of the PolitiFact website until end of April 2018. We observe that the quantity of facts varies significantly across years. Also, we observed that the rate of true/false facts is in favour for false facts for almost every year.

Snippets: For each fact, we fetch associated snippets by using the entire fact as a query to the Bing search engine. Furthermore, for each fact, we are interested in analysing the evolution of the associated snippets over time. This is why we associate a time window to each fact, specifically a ten days long time window occurring just before the day the fact was fact-checked on PolitiFact.

The idea is that false facts tend to arouse much more interest in a smaller time window than true facts, therefore we expect that false facts will have more snippets in a smaller time window than true facts.

For each fact, for each day in the fact's time window, we perform a Bing query and we fetch the first ten pages of results at most with a maximum of ten snippets per page.

On figure A.2, we see a screenshot of a Bing page of snippet results with the associated fact and the associated day from the fact's time window. On figure A.3, we see a screenshot of a Bing page of results with page numbers.

On figure 5.3, we observe that true and false facts have similar numbers of associated snippets. This undermines the hypothesis that false facts arouse more interest and hence more Bing results than true facts.

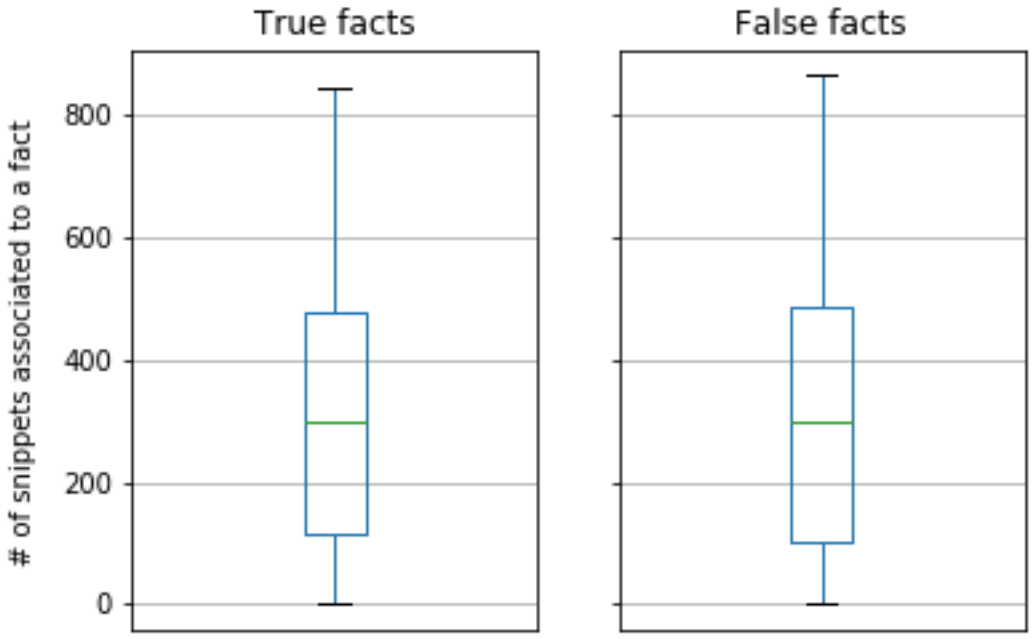

Figure 5.3: Distribution of numbers of snippets per true and false facts.

Documents: After the relevant snippet discovery step described in Section 4.2, we fetch relevant documents associated to relevant snippets by fetching web pages using HTTP requests and extracting text content from these web pages[4].

Surprisingly, we succeed at retrieving all documents associated to relevant snippets.

Data selection

Snippets: A first problem we encounter with snippets is that some snippets come from PolitiFact. We can see on figure 5.4 the number of snippets from PolitiFact. On figure 5.5, we observe that the distribution of snippets from Politifact is similar between true and false facts.

We consider this as a problem because the documents associated to these snippets contain ground truth labels related to facts, therefore a baseline model for binary classification can predict without error the label associated to a fact by looking at the associated document from PolitiFact. To solve this problem, we simply delete snippets from PolitiFact.

Another problem is that some snippets are not in english. To tackle this problem, we use an existing dictionary of most frequent words in english[5] and we associate to each snippet the number of occurrences of frequent english words in this snippet. We observe

[4]https://www.quora.com/How-can-I-extract-only-text-data-from-HTML-pages
[5]https://en.wikipedia.org/wiki/Most_common_words_in_English

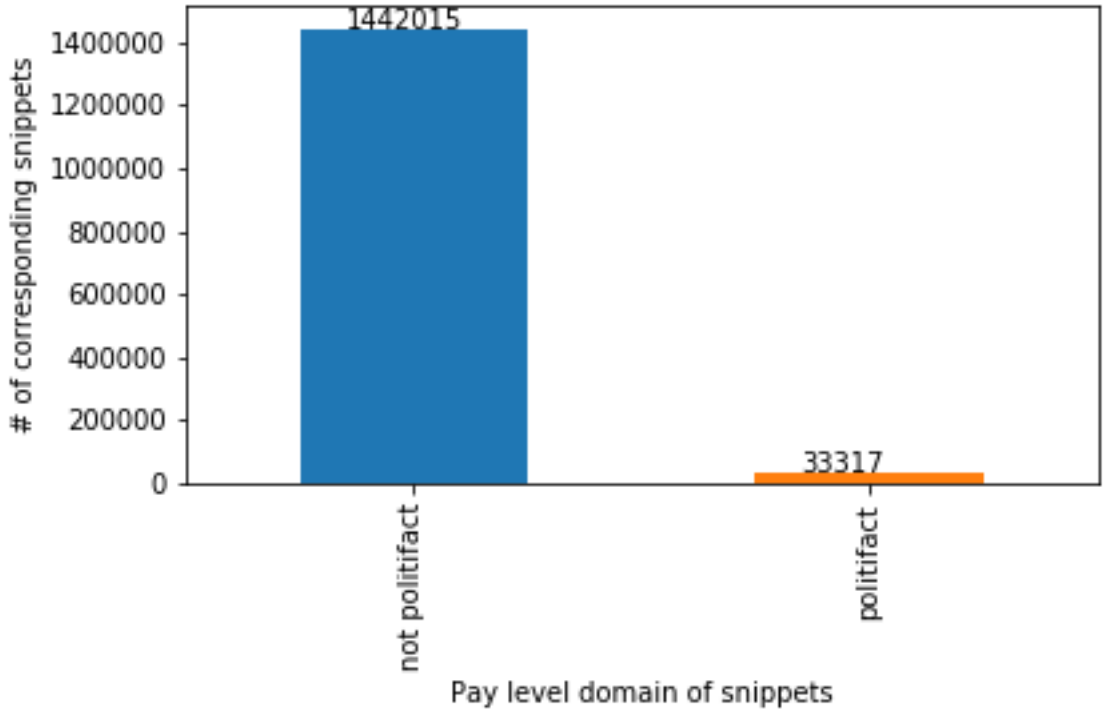

Figure 5.4: Quantity of snippets with an url from PolitiFact.

on figure 5.6 the distribution of these numbers of occurrences among snippets. On figure 5.7, we observe similar distributions on average number of occurrences of snippets for true and false facts. We simply delete all snippets containing 0 frequent words in english.

Documents: After fetching relevant documents associated to facts, we observe in figure 5.8 that half of our facts have a number of associated documents less or equal than 2. In order to normalise our data, we delete facts with less than 2 relevant documents associated and we select for each remaining fact the 2 associated relevant documents with the best similarity scores.

To facilitate our future data processing steps, we aggregate each couple of relevant documents associated to a same fact into a single document by concatenating their content.

Data transformation

Prior standard preprocessing: We apply the following preprocessing steps on facts, snippets and documents:

- *Lowercase transformation:* We change uppercase characters in text with their lowercase character associated.
- *Tokenization:* We segment texts in list of words.

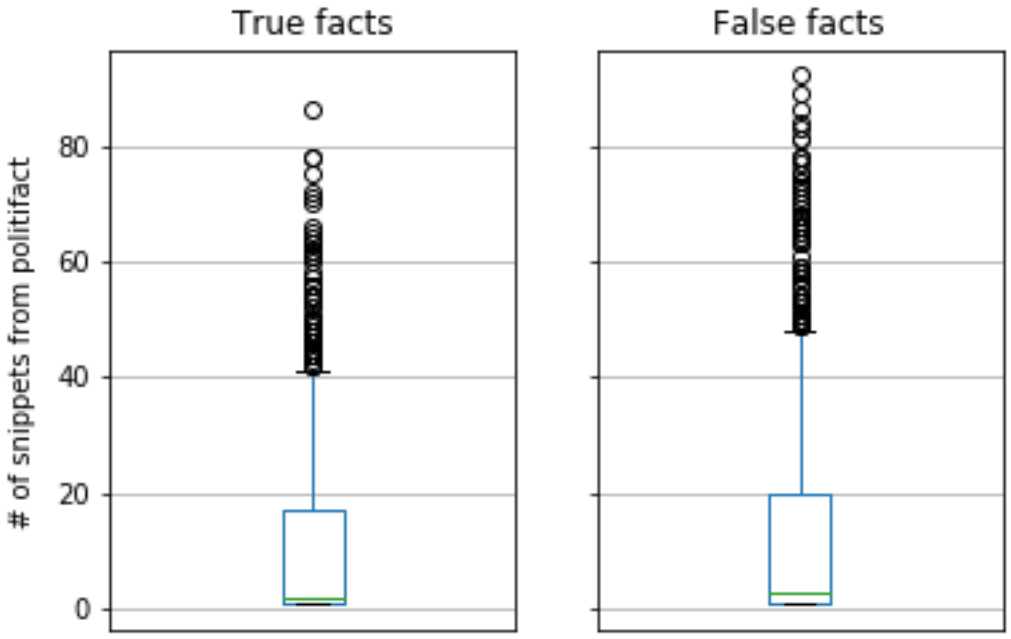

Figure 5.5: Distribution of numbers of snippets from PolitiFact per true and false facts.

- *Part of speech tagging:* We annotate each word with a part of speech tag. A part of speech tag refers to a category of words sharing the same grammatical properties.
- *Select interesting part of speech tags:* We keep only nouns, verbs, adverbs and adjectives.

Original preprocessing We designed original preprocessing steps described below:

- *Simple quotes preprocessing:* Different UTF-8 characters exists for simple quotes. We replace all these different characters by one character for simple quotes.
- *Suppress noise characters:* Some characters introduce noise in our data such as emoticons or problems due to encoding format.
- *Replace abbreviations:* Many abbreviations exists for name of countries or organisations (e.g. u.s.a., u.k., n.a.t.o., etc.)
- *Replace separation characters:* Instead of being separated by spaces, words are sometimes joined with other characters (e.g. '/', '-', etc.). We replace these characters with spaces.

After these preprocessing steps described above, we filter words using a simple regular expression to match english words:

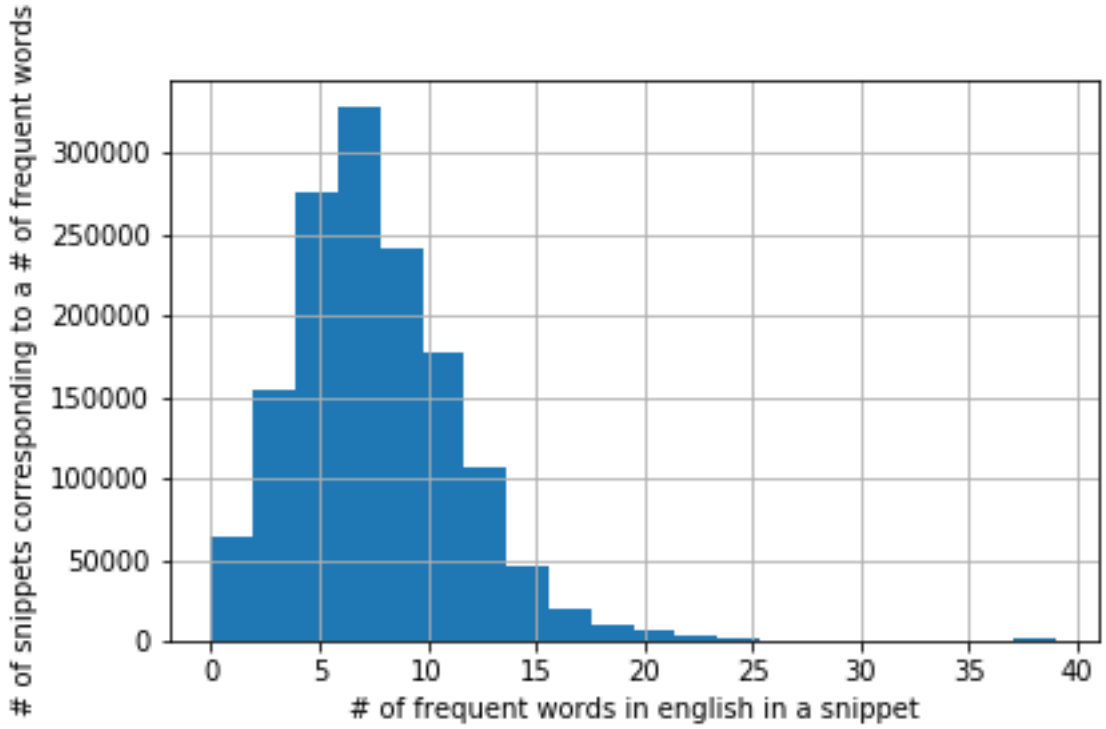

Figure 5.6: Distribution of numbers of frequent english words in snippets.

Definition 10. *English words regular expression:*
A regular expression that matches only words containing only ascii letters and the simple quote character.

We design all these original preprocessing steps by analysing snippets. We apply these processing steps on facts, snippets and documents.

Post-processing We apply the following post-processing steps over facts, snippets and documents:

- *Stop words detection:* Stop words are words frequently used but without important meaning (e.g. 'say', 'have', 'people', etc.). We remove stop words from data.
- *Phrase detection:* Phrases are couple of words which frequently appear side by side. We transform data into list of words an phrases.

Dictionary configurations For each step of our approach, we build corresponding dictionaries.

- *For relevant snippet discovery:*

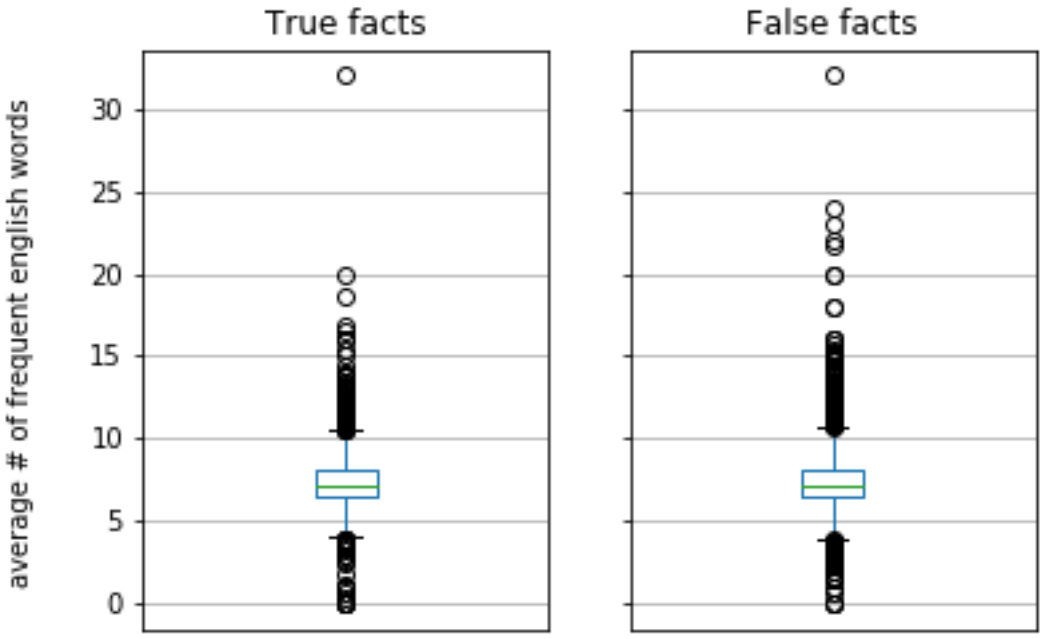

Figure 5.7: Distribution of average numbers of frequent english words in snippets per true and false facts.

  - We build a first dictionary from the collection of snippets. We filter out words or phrases contained in less than 5 snippets or contained in more than half of all the snippets.
  - We build 6 dictionaries by keeping only the most frequent words or phrases from the prior dictionary. The number of words and phrases varies from 10 000 to 35 000 with a step of 5 000 words and phrases.

- *For binary classification:*

  - We build a first dictionary from the collection of relevant documents. We filter out words or phrases contained in less than 5 documents or contained in more than half of all the documents.
  - We build 10 dictionaries by keeping only the most frequent words or phrases from the prior dictionary. The number of words and phrases varies from 2 000 to 20 000 with a step of 2 000 words and phrases.
  - We build a second dictionary from the collection of facts. We filter out words or phrases contained in less than 5 facts or contained in more than half of all the facts.

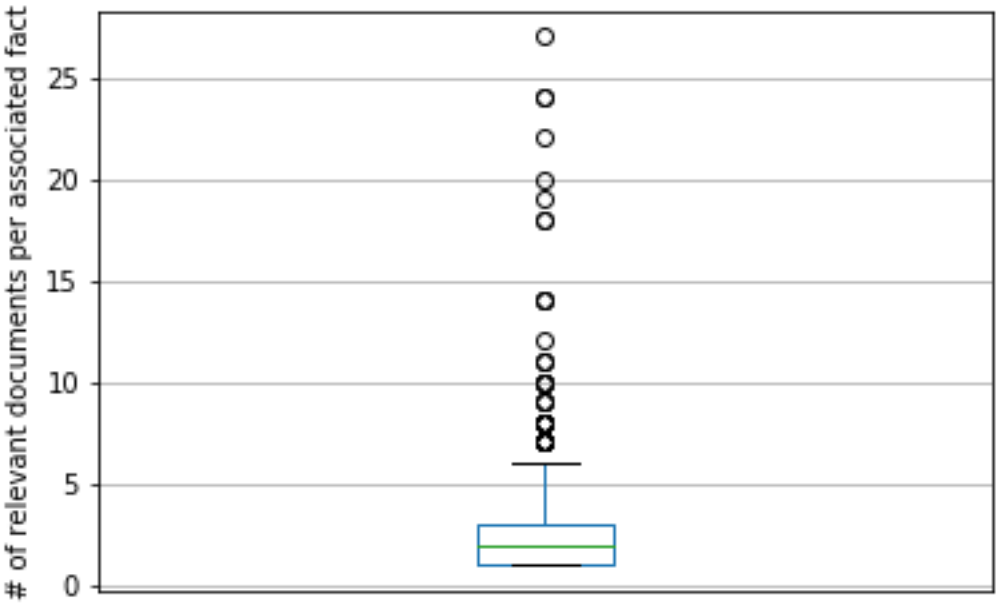

Figure 5.8: Distribution of numbers of relevant documents per associated fact.

– We build 10 dictionaries by keeping only the most frequent words or phrases from the prior dictionary. The number of words and phrases varies from 90 to 180 words with a step of 10 words and phrases.

5.2 Ground Truth and Metrics

Each step of our approach is evaluated using specific ground truth data and metrics that we introduce below.

Relevant snippet discovery

Ground truth: We want a collection of snippets labelled as relevant or irrelevant by an expert. The author of this work acts as an expert and builds the ground truth data as follows:

- We count for each fact and for each day in the time window associated to this fact the number of corresponding snippets.
- We select the 10 couples of facts and days with the maximum numbers of snippets associated. The idea is that we want few facts to facilitate the process of labelling but we want a large number of snippets associated to have a reasonable amount of ground truth data. However, facts have a large number of snippets associated as we see in figure 5.5, therefore we decide to restrict the number of snippets by selecting one day in the time window associated to the fact.
- For each snippet, we compare the content of the snippet with the associated fact and we decide whether or not this fact is relevant.
- We can see in figure 5.9 the distribution of relevant and irrelevant snippets in our ground truth data. The superior number of irrelevant snippets suggests that the majority of results from a Bing search engine query are irrelevant.

Metrics: We need a metric to discriminate between features. A good feature used in a combination with a similarity measure provides high similarity scores to relevant snippets and low similarity scores to irrelevant ones. Here are our expectations sorted by decreasing values of importance for a good metric:

- We want to identify relevant snippets with 100% precision. In other words, we prefer the quality of snippets over quantity. Irrelevant snippets can have a negative impact for the binary classification experiment.
- We want to have a maximum number of relevant snippets associated to facts.

For all the reasons explained above, we choose the following metric.

Definition 11. *Relevant document discovery metric : Given a feature and the corresponding similarity scores associated to the snippets, we sort the snippets by their similarity scores in decreasing order and we count the top number of relevant snippets.*

Binary classification

Ground truth: Our ground truth data is the collection of true and false facts with aggregated relevant documents associated described in section 5.1.

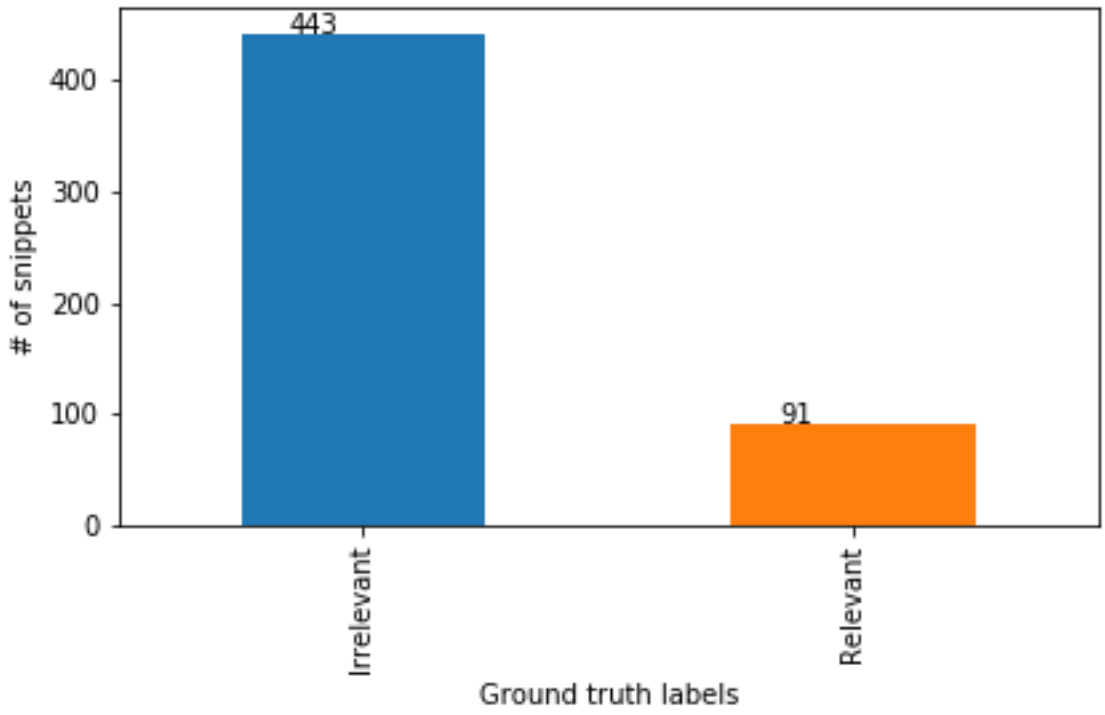

Figure 5.9: Distribution of relevant and irrelevant snippets in the ground truth data for relevance snippet discovery.

Metrics: We need a metric to discriminate between binary classifiers with associated features. A good combination of a binary classifier and a feature predicts true labels for true facts and false labels for false facts. Here are our expectations of equal importance for a good metric:

- We want a high percentage of true facts among facts predicted as true. In other words, we want high precision score.
- We want to predict a high percentage of true labels among true facts, that is high recall score.

Considering our expectations, we choose the following metric:

Definition 12. *Binary classification metric: Given a couple of binary classifier and feature, and predicted labels associated to true and false facts. We choose the harmonic mean of precision and recall called F-measure.*

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall}$$

5.3 Configuration and Baselines

Using ground truth data and metrics described in section 5.2, we evaluate different feature configurations of our data and compare their performances against different baselines. We detail these configurations and baselines for each step of our experiment below.

Relevant document discovery

Configuration: We build topic modelling representations of our data by varying different parameters:

- *Size of the dictionary:* We vary the number of words in the dictionary from 10 000 to 35 000 with a step of 5 000 as we described in subsection 5.1.
- *Topic modelling technique:* We choose different topic modelling techniques, namely Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA) and Random Projection (RP).
- *Number of topics:* We configure the topic modelling techniques with different numbers of topics starting from 100 to 300 with a step of 50.

Overall, we have 90 different topic modelling representations.

For each topic modelling representation and each baseline, we associated corresponding similarity scores to snippets in ground truth data and we apply the relevant snippet discovery metric on these similarity measures.

Baselines: Using the ground truth data and the metric described in subsection 5.2, we evaluate and compare performances between the above topic modelling representations and a set of baselines. These baselines are representations of our data and we build them by varying different parameters:

- *Size of the dictionary:* As we do with topic modelling representations, we vary the number of words in the dictionary from 10 000 to 35 000 with a step of 5 000.
- *Representation technique:* We use both BOW and TF-IDF representation techniques.

Overall, we have 12 baselines.

Binary classification

Configuration: We build topic modelling representations of our facts and aggregated relevant documents by varying different parameters:

- *Size of the dictionary:* We vary the number of words in the dictionary from 90 to 180 with a step of 10 for facts and from 2 000 to 18 000 with as step of 2 000 for aggregated relevant documents.

- *Topic modelling technique:* We choose different topic modelling techniques, namely Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA) and Random Projection (RP) for both facts and aggregated relevant documents.

- *Number of topics:* We configure the topic modelling techniques with different numbers of topics starting from 10 to 50 with a step of 50 for facts and from 100 to 300 with a step of 40 for aggregated relevant documents.

Overall, we have 150 different topic modelling representations for facts and 162 for aggregated relevant document respectively.

We can't train and test a standard binary classifier on all of these topic modelling representations because it would be too much time consuming.

To reduce the number of topic modelling representations, we first apply a feature selection step to select the best topic modelling representation for facts and for aggregated relevant documents.

Specifically, for each topic modelling representation and each baseline, we train and test a logistic regression classifier using grid search hyper parameter optimisation and 10-fold cross validation. Then, we check that topic modelling representations have better performances than baselines and we select the best topic modelling representation for facts and aggregated relevant documents respectively.

Once we finish the feature selection step, for each best topic modelling representation of facts and aggregated relevant documents respectively, we train and test a support vector machine classifier using grid search hyper parameter optimisation and 10-fold cross validation. Finally, we check that the best topic modelling representation of aggregated relevant documents has superior performances over the best topic modelling representation of facts.

Baselines: Our binary classification baselines are representations of facts and aggregated relevant documents respectively. Here are the different parameters used to build these baselines:

- *Size of the dictionary:* As we do with topic modelling representations, we vary the number of words in the dictionary from 90 to 180 with a step of 10 for facts and from 2 000 to 18 000 with a step of 2 000 for aggregated relevant documents.

- *Representation technique:* We use both BOW and TF-IDF representation techniques for facts and aggregated relevant documents.

Overall, we have 20 different baselines for facts and 18 for aggregated relevant documents respectively.

6

Evaluation Results

In this chapter, for each step of our experiment, we first introduce the results, then we discuss the potential of our experiment and approach and finally we enumerate the limitations.

6.1 Results

In this section, we introduce the results of our experiment for relevant snippet discovery and binary classification.

Relevant Snippet Discovery

We see in table 6.1 that the baseline for relevant snippet discovery with the highest score given by the relevant snippet discovery metric is the BOW representation with the associated dictionary of 10 000 words and with the score of 4.

On the other hand, we see in table 6.2 that the topic modelling representation with the highest score given by the relevant snippet discovery metric is the LDA representation with the associated dictionary of 30 000 words and with the score of 7.

Therefore, we choose the LDA representation with the associated dictionary of 10 000 words to select relevant snippets. We choose a similarity threshold of 0.65, hence each snippet with a similarity score to its associated fact above or equal to 0.65 is considered as relevant whereas each ones with similarity scores below 0.65 are considered as irrelevant and we delete them. We can see on figure 6.1 that 0.65 is the minimum value above all similarity scores of irrelevant snippets.

Table 6.1: Relevant snippet discovery metric scores for baseline representations with associated sizes of dictionaries from 10 000 to 35 000.

	10K	**15K**	**20K**	**25K**	**30K**	**35K**
BOW	**4**	3	2	2	2	2
TF-IDF	1	1	1	1	1	1

Table 6.2: Relevant snippet discovery metric scores for topic modelling representations with associated sizes of dictionaries from 10 000 to 35 000 and associated number of topics from 100 to 300.

	LSI					**LDA**					**RP**				
	100	**150**	**200**	**250**	**300**	**100**	**150**	**200**	**250**	**300**	**100**	**150**	**200**	**250**	**200**
10K	0	1	1	0	0	0	5	4	2	6	3	3	5	2	6
15K	1	1	1	1	0	0	4	3	5	3	2	3	4	2	6
20K	1	1	1	1	0	6	2	1	1	1	2	3	1	3	4
25K	0	1	1	1	0	0	3	5	1	5	4	4	4	4	3
30K	0	1	1	1	1	3	0	**7**	1	5	6	3	4	1	2
35K	0	1	1	1	0	2	2	2	6	1	4	2	3	5	1

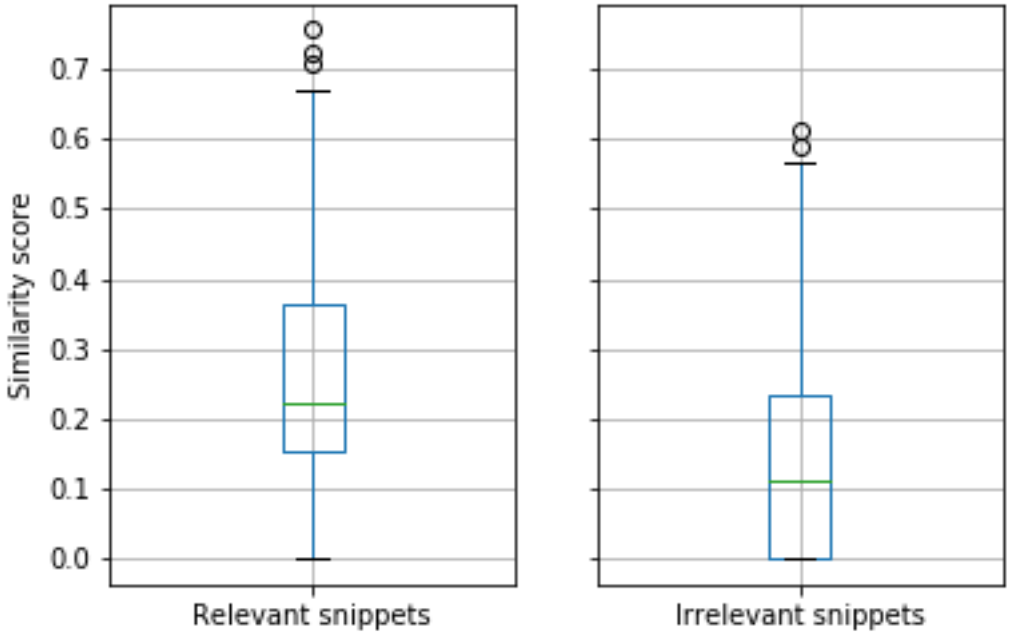

Figure 6.1: Distribution of similarity scores among relevant and irrelevant snippets with a dictionary size of 30K and 200 topics with LDA topic modeling.

Table 6.3: Results from binary classification experiment using BOW and TF-IDF representations of facts with size of dictionaries from 100 to 300 words.

	90	**100**	**110**	**120**	**130**	**140**	**150**	**160**	**170**	**180**
BOW	0.61	0.57	0.57	0.56	0.58	0.58	0.58	0.60	0.61	0.60
TF-IDF	**0.62**	0.60	0.60	0.59	0.60	0.59	0.58	0.61	0.60	0.60

Table 6.4: F-measure scores from binary classification experiment using BOW and TF-IDF representations of aggregated relevant documents with size of dictionaries from 2000 to 18000 words.

	2K	**4K**	**6K**	**8K**	**10K**	**12K**	**14K**	**16K**	**18K**
BOW	0.53	0.48	0.48	0.48	0.47	0.46	0.48	0.47	0.48
TF-IDF	0.57	0.50	0.53	0.46	0.53	0.46	**0.60**	0.47	0.47

Binary classification

We see in table 6.3 and in table 6.4 the F-measure scores associated to binary classification baselines for facts and aggregated relevant documents respectively.

In table 6.5, we see that the topic modelling representation of facts with the highest f-measure score of 0.67 outperforms the baseline representation of facts with the highest score of 0.62.

Similarly, in table 6.6, we see that topic modelling representations of aggregated relevant documents with the highest f-measure score of 0.66 outperform the baseline representation of aggregated relevant documents with the highest score of 0.60.

Finally, in table 6.7, we observe that we get superior performances for selected fact representation over selected aggregated relevant document representations.

6.2 Potential

In this section, we discuss the potential performances of the relevant snippet discovery and binary classification steps in the light of the results described in section 6.1.

Relevant Document Discovery

First, we demonstrate in our results that choosing the right parameters for the dictionary and topic modelling configurations has a great impact on relevant snippet discovery performances. Indeed, using the same topic modelling technique such as LDA, the scores vary from 0 to 7 depending on the size of the dictionary and the number of topics.

Another type of technique that can improves relevant snippet discovery performances is topic modelling techniques since we demonstrate an improvement of performances of 75% using topic modelling techniques compared to standard text-mining techniques such as BOW and TF-IDF.

However, when we build the ground truth data, we observe that most of the snippets fetched from the Bing search engine are irrelevant with regards to the associated facts.

Table 6.5: F-measure scores from binary classification experiment using LSI, LDA and RP representations of facts with size of dictionaries from 90 to 180 words and number of topics from 10 to 50.

		90	**100**	**110**	**120**	**130**	**140**	**150**	**160**	**170**	**180**
LDA	**10**	0.59	**0.67**	0.64	0.57	0.55	0.59	0.53	0.58	0.51	0.59
	20	0.62	0.62	0.58	0.62	0.55	0.53	0.60	0.59	0.61	0.60
	30	0.62	0.65	0.62	0.55	0.50	0.59	0.61	0.59	0.63	0.65
	40	0.56	0.62	0.52	0.59	0.62	0.53	0.52	0.54	0.61	0.61
	50	0.64	0.52	0.62	0.53	0.58	0.53	0.49	0.51	0.54	0.57
LSI	**10**	0.64	0.61	0.61	0.60	0.61	0.61	0.61	0.64	0.61	0.62
	20	0.65	0.63	0.64	0.64	0.64	0.64	0.64	0.62	0.63	0.61
	30	0.63	0.63	0.65	0.62	0.61	0.63	0.63	0.62	0.60	0.62
	40	0.63	0.62	0.63	0.63	0.62	0.63	0.62	0.60	0.59	0.60
	50	0.63	0.62	0.64	0.62	0.63	0.61	0.60	0.59	0.59	0.59
RP	**10**	0.58	0.55	0.55	0.57	0.50	0.51	0.52	0.52	0.60	0.50
	20	0.58	0.50	0.59	0.56	0.53	0.61	0.58	0.60	0.60	0.56
	30	0.53	0.56	0.54	0.63	0.55	0.50	0.55	0.52	0.56	0.53
	40	0.61	0.44	0.59	0.57	0.55	0.58	0.54	0.57	0.55	0.50
	50	0.57	0.59	0.57	0.66	0.58	0.60	0.56	0.53	0.62	0.54

Table 6.6: F-measure scores from binary classification experiment using LSI, LDA and RP representations of documents with size of dictionaries from 2000 to 18000 words and number of topics from 100 to 300.

		2K	**4K**	**6K**	**8K**	**10K**	**12K**	**14K**	**16K**	**18K**
LDA	**100**	0.64	0.57	0.54	0.55	0.57	0.48	0.55	0.49	0.46
	140	0.55	0.47	0.49	0.47	0.53	0.58	0.44	0.56	0.50
	180	0.60	0.48	0.53	0.56	0.48	0.53	0.57	0.47	0.58
	220	0.54	0.55	0.53	0.62	0.60	0.58	0.53	0.58	0.57
	260	0.49	0.60	0.46	0.47	0.58	0.52	0.55	0.47	0.52
	300	0.46	0.52	0.54	0.58	**0.66**	0.55	0.53	0.47	0.47
LSI	**100**	0.51	0.53	0.60	0.49	0.52	0.54	0.52	0.55	0.53
	140	0.53	0.52	0.52	0.55	0.53	0.54	0.49	0.51	0.49
	180	0.51	0.53	**0.66**	0.59	0.51	0.54	0.55	0.50	0.52
	220	0.51	0.63	0.59	0.53	0.51	0.54	0.53	0.50	0.48
	260	0.52	0.50	0.51	0.46	0.54	0.60	0.52	0.53	0.48
	300	0.51	0.53	0.53	0.46	0.54	0.54	0.53	0.50	0.52
RP	**100**	0.51	0.54	0.53	0.54	0.53	0.51	0.50	0.54	0.52
	140	0.51	0.54	0.50	0.51	0.54	0.50	0.48	0.51	0.51
	180	0.50	0.49	0.52	0.53	0.46	0.53	0.52	0.55	0.52
	220	0.51	0.47	0.50	0.54	0.50	0.54	0.44	0.51	0.52
	260	0.53	0.46	0.48	0.53	0.56	0.53	0.50	0.54	0.50
	300	0.52	0.46	0.53	0.48	0.52	0.47	0.46	0.54	0.51

Table 6.7: F-measure scores from binary classification experiment using the selected topic modelling representations of facts and aggregated relevant documents.

Facts	**Aggregated relevant documents**	**Aggregated relevant document**
LDA 100 words 10 topics	**LDA 10K words 300 topics**	**LSI 6K words 180 topics**
0.64	0.58	0.51

Binary classification

Our experiment fails at demonstrating superior performances for aggregated relevant document representations against fact representation.

However, we demonstrate superior performances for topic modelling representations against standard text-mining representations with an improvement of performances of 7% for facts and 10% for aggregated relevant documents respectively.

6.3 Limitations

Our approach and experiment suffer several limitations. In this section, we discuss possible limitations associated with our data, our relevant snippet discovery and binary classification steps.

Data

Our data suffers several limitations:

- We collect data from only one fact-checking website. Our data will gain in size if we collect data from different fact-checking websites. Moreover, it can gain reliability if we identify similar facts discussed across different fact-checking websites and if we compare all veracity verdicts associated to the same fact across these websites.
- We collect snippets from Bing search engine queries with time window parameters but we observe in figure A.2 that Bing does not always respect the time information of our queries. A preliminary analysis of Bing results reliability could help us to better use time windows with the Bing search engine.
- We feed the Bing search engine with the entire content of the facts. Other approaches exists to build Bing search queries such as extracting keywords from the fact.
- Maybe the Bing search engine adapts its results based on the user's profile.
- Data preprocessing steps can have a great impact on performances for relevant snippets discovery and binary classification. However, we apply many data preprocessing steps without evaluating them.
- For each fact, we aggregate the top 2 relevant documents associated. We can try to aggregate the top k relevant documents using different values of k and evaluate the performance of a standard representation of facts and snippets for relevant snippets discovery for each possible value of k.

Relevant snippet discovery

Our relevant snippet discovery step has many limitations:

- Our ground truth data is small and may contains errors. A solution to get bigger and more reliable ground truth data is to use many experts to label the data and keep only the data with a high agreement score between experts.
- The metric we use for relevant snippet discovery does not consider association between snippets and facts. This is a problem because a relevant snippet discovery representation can discriminate well between relevant and irrelevant snippets for a set of facts and not discriminate well for the other facts. We want a relevant snippet discovery representation that discriminate with relatively equal performances for each fact, therefore our data for binary classification will be homogenised.
- We use cosinus distance as similarity measure but other measures exist. Furthermore, we can apply binary classifier to label snippets as relevant or irrelevant.
- Our approach does not take into account interesting features for relevant snippet discovery such as entities, dates, locations, etc.

Binary classification

Finally, we introduce the limitations of our binary classification step:

- Our representations of relevant documents does not take into account the time information associated with documents.
- We do not look at the evolution of numbers of Bing results associated to true and false facts over time. Our intuition is that false facts tend to have a bigger number of results in a smaller time window than true facts.
- The number of our configurations is the cardinal product between all possible dictionary configurations and topic modelling configurations. To reduce this number of configurations, we can first select the best dictionary configuration by training and testing a binary classifier with a standard text-mining representation techniques and then evaluate all the topic modelling configurations associated with the best dictionary configuration.

7

Conclusion and Future Work

We have introduced our approach for automatic fact-checking based on two main steps : relevant snippet discovery and binary classification. We have built a data set using the fact-checking website PolitiFact and the Bing search engine.

Evaluation results do not suggest superior performances for aggregated relevant document representations over fact representations.

While we did our experiments on data fetched from PolitiFact, we will consider to test our approach on data from other fact-checking websites.

Another direction for further research is to find similar facts across fact-checking websites and analyse agreement between fact-checking websites annotations and build a unified data set.

Furthermore, we can evaluate each preprocessing step in the same way that we evaluate features. For binary classification, we will evaluate each preprocessing step by applying logistic regression on both bow representations of preprocessed facts and aggregated relevant documents and bow representations of not preprocessed facts and aggregated relevant documents.

Finally, we need to evaluate which size of the dictionary to use before we do our experiments. This would reduce the number of parameters in the configurations of our experiments and allow us to test more values for each parameter.

A

Figure A.1: Screenshot of the PolitiFact web page.

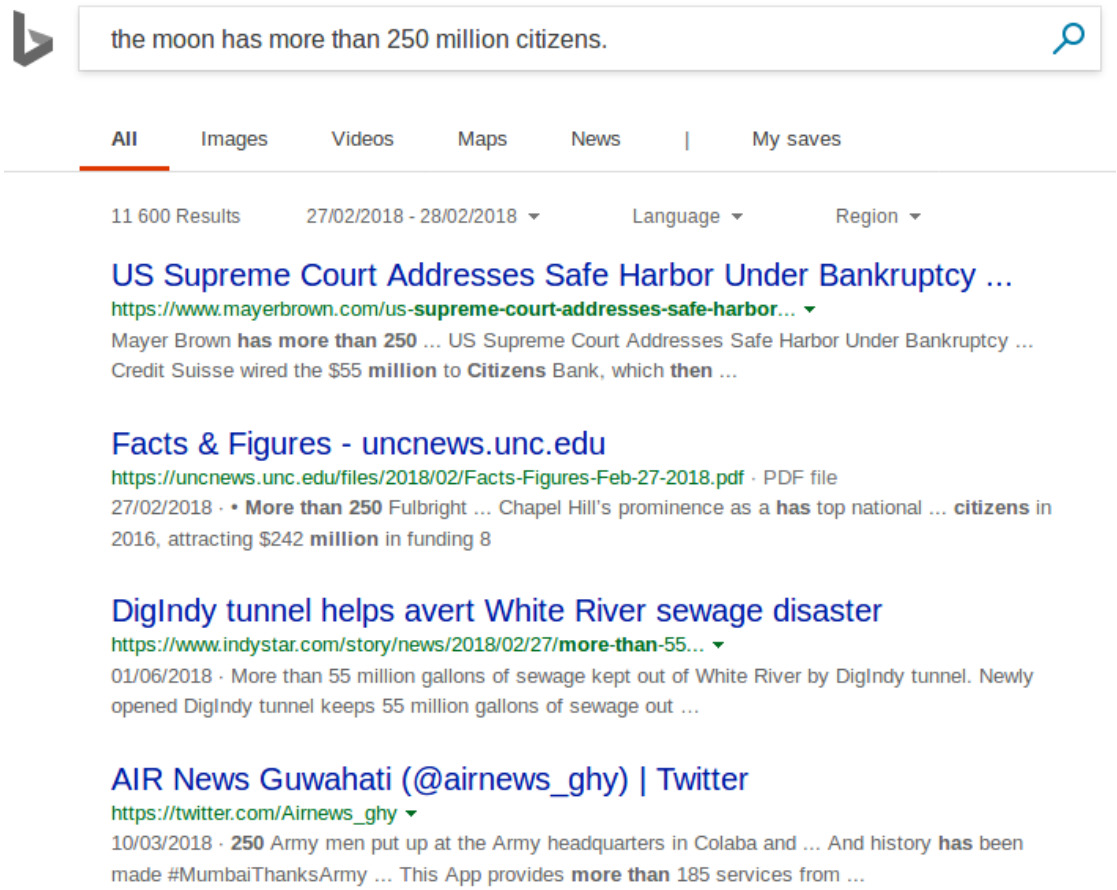

Figure A.2: Screenshot of a Bing results page with associated query content and time window.

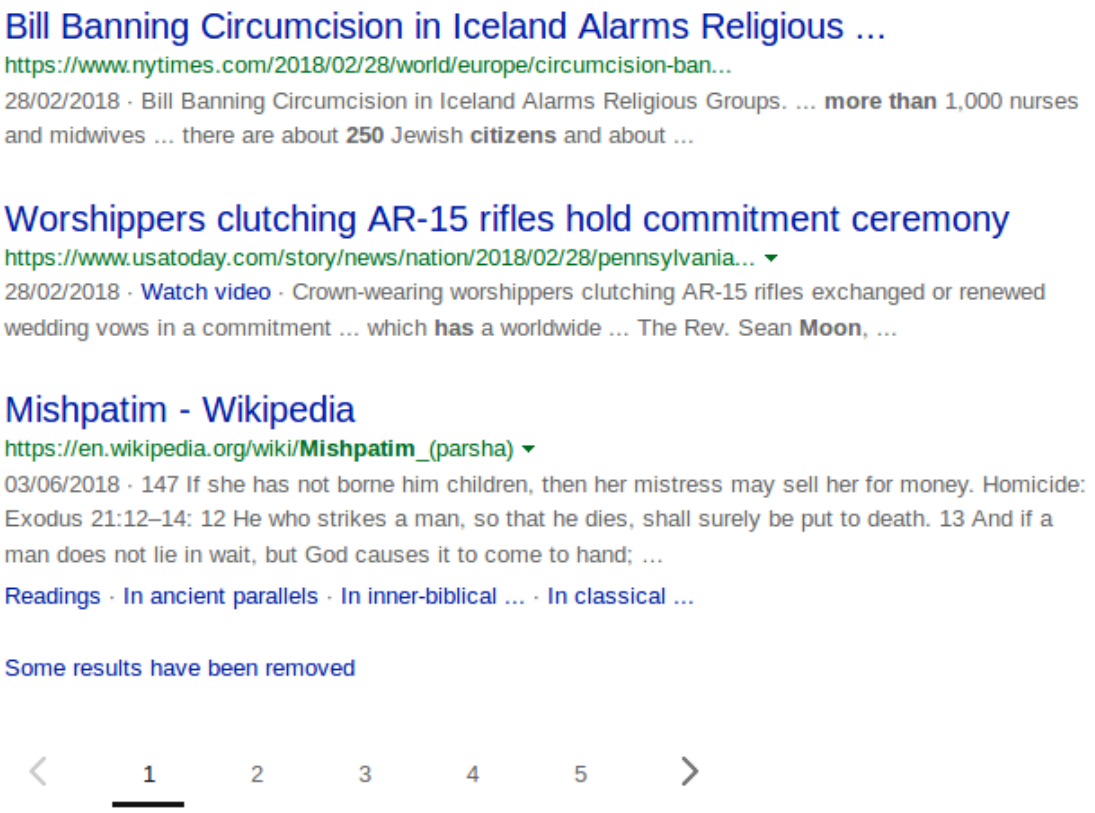

Figure A.3: Screenshot of a Bing results page with associated number of pages of results.

Bibliography

[1] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wiko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, Wei Zhang. "*Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion.*" in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601-610, ACM, 2014

[2] Ran Yu, Ujwal Gadiraju, Besnik Fetahu, Olivier Lehmberg, Dominique Ritze and Stefan Dietze. "*KnowMore - Knowledge Base Augmentation with Structured Web Markup.*", in *Semantic Web Journal*, 2018.

[3] Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, Cong Yu. "*The Quest to Automate Fact-Checking.*" in *Proceedings of Computation + Journalism Symposium*, 2015.

[4] William Yang Wang. '*"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection.*', in *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pp. 422–426, 2017.

[5] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen and Gerhard Weikum. '*Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media.*', in *Proceedings of the 26th International World Wide Web Companion*, pp. 1003-1012, 2017.

[6] Xuezhi Wang, Cong Yu, Simon Baumgartner and Flip Korn. '*Relevant Document Discovery for Fact-Checking Articles.*', in *Companion Proceedings of the Web Conference*, pp. 525-533, 2018.

[7] Soroush Vosoughi, Deb Roy and Sinan Aral. '*The spread of true and false news online.*', in *Science 359*, pp. 1146–1151, 2018.