

**METHODS FOR IMPROVING ENTITY LINKING AND  
EXPLOITING SOCIAL MEDIA MESSAGES ACROSS CRISES**

Von der Fakultät für Elektrotechnik und Informatik  
der Gottfried Wilhelm Leibniz Universität Hannover  
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

**Dr. rer. nat.**

genehmigte Dissertation  
von

**M.Sc. Renato Stoffalette Joao**

geboren am 20. Oktober 1984 in Osvaldo Cruz/SP - Brasilien

Hannover, Deutschland, 2023

**Referent: Prof. Dr. techn. Wolfgang Nejd**  
**Korreferent: Prof. Dr. Stefan Dietze**  
**Tag der Promotion: 21.03.2023**

## ABSTRACT

Entity Linking (EL) is the task of automatically identifying entity mentions in texts and resolving them to a corresponding entity in a reference knowledge base (KB). There is a large number of tools available for different types of documents and domains, however the literature in entity linking has shown the quality of a tool varies across different corpus and depends on specific characteristics of the corpus it is applied to. Moreover the lack of precision on particularly ambiguous mentions often spoils the usefulness of automated disambiguation results in real world applications.

In the first part of this thesis I explore an approximation of the difficulty to link entity mentions and frame it as a supervised classification task. Classifying difficult to disambiguate entity mentions can facilitate identifying critical cases as part of a semi-automated system, while detecting latent corpus characteristics that affect the entity linking performance. Moreover, despite the large number of entity linking tools that have been proposed throughout the past years, some tools work better on short mentions while others perform better when there is more contextual information. To this end, I proposed a solution by exploiting results from distinct entity linking tools on the same corpus by leveraging their individual strengths on a per-mention basis. The proposed solution demonstrated to be effective and outperformed the individual entity systems employed in a series of experiments.

An important component in the majority of the entity linking tools is the probability that a mentions links to one entity in a reference knowledge base, and the computation of this probability is usually done over a static snapshot of a reference KB. However, an entity's popularity is temporally sensitive and may change due to short term events. Moreover, these changes might be then reflected in a KB and EL tools can produce different results for a given mention at different times. I investigated the prior probability change over time and the overall disambiguation performance using different KB from different time periods.

The second part of this thesis is mainly concerned with short texts. Social media has become an integral part of the modern society. Twitter, for instance, is one of the most popular social media platforms around the world that enables people to share their opinions and post short messages about any subject on a daily basis. At first I presented one approach to identifying informative messages during catastrophic events using deep learning techniques. By automatically detecting informative messages posted by users during major events, it can enable professionals involved in crisis management to better estimate damages with only relevant information posted on social media channels, as well as to act immediately. Moreover I have also performed an analysis study on Twitter messages posted during the Covid-19 pandemic. Initially I collected 4 million tweets posted in Portuguese since the beginning of the pandemic and provided an analysis of the debate around the pandemic. I used topic modeling, sentiment analysis and hashtags recommendation techniques to provide insights around the online discussion of the Covid-19 pandemic.

**Keywords:** *Entity Linking, Ensemble Learning, Knowledge Base, Deep Learning*

## ZUSAMMENFASSUNG

Entity Linking (EL) ist die Aufgabe, automatisch Entitätserwähnungen in Texten zu identifizieren und sie zu einer entsprechenden Entität in einer Referenz-Wissensbasis (WB) zu verlinken. Es gibt eine große Anzahl von Tools für verschiedene Arten von Dokumenten und Domänen. Die Literatur zu Entity Linking hat jedoch gezeigt, dass die Qualität eines Tools je nach Korpus variiert und von den spezifischen Eigenschaften des Korpus abhängt. Darüber hinaus beeinträchtigt die mangelnde Präzision bei besonders mehrdeutigen Erwähnungen oft die Nützlichkeit der automatischen Disambiguierungsergebnisse in realen Anwendungen.

Im ersten Teil dieser Arbeit untersuche ich einen Ansatz für die Problemstellung, Entitätserwähnungen zu verknüpfen, und stelle sie als überwachte Klassifizierungsaufgabe dar. Die Klassifizierung von schwer zu disambiguierenden Entitätserwähnungen kann die Identifizierung von kritischen Fällen als Teil eines halbautomatischen Systems erleichtern und gleichzeitig latente Korpus-Charakteristika aufdecken, die die Entitätsverknüpfungsleistung beeinflussen. Trotz der großen Anzahl von Tools zur Verknüpfung von Entitäten, die in den letzten Jahren vorgeschlagen wurden, funktionieren einige Tools besser bei kurzen Erwähnungen, während andere besser funktionieren, wenn mehr Kontextinformationen vorhanden sind. Zu diesem Zweck habe ich eine Lösung vorgeschlagen, bei der die Ergebnisse verschiedener Entity-Linking-Systeme auf demselben Korpus genutzt werden, indem ihre individuellen Stärken je nach Erwähnung genutzt werden. Die vorgeschlagene Lösung erwies sich als effektiv und übertraf die einzelnen Entity-Linking-Systeme in einer Reihe von Experimenten.

Eine wichtige Komponente der meisten Entity-Linking-Systeme ist die Wahrscheinlichkeit, dass eine Erwähnung auf eine Entität in einer Referenz-Wissensbasis verweist, und die Berechnung dieser Wahrscheinlichkeit erfolgt in der Regel über eine statische Momentaufnahme der WB. Die Popularität einer Entität ist jedoch zeitabhängig und kann sich aufgrund von kurzfristigen Ereignissen ändern. Außerdem können sich diese Änderungen in einer WB widerspiegeln, und EL-Tools können für eine bestimmte Erwähnung zu verschiedenen Zeiten unterschiedliche Ergebnisse liefern. Ich untersuchte die Veränderung der Wahrscheinlichkeit im Laufe der Zeit und die allgemeine Qualität der Disambiguierung anhand verschiedener WB aus unterschiedlichen Zeiträumen.

Der zweite Teil dieser Arbeit befasst sich hauptsächlich mit kurzen Texten. Soziale Medien sind zu einem festen Bestandteil der modernen Gesellschaft geworden. Twitter als eine der beliebtesten globalen Social-Media-Plattformen ermöglicht es den Menschen, ihre Meinung mitzuteilen und täglich Kurznachrichten zu beliebigen Themen zu veröffentlichen. Zunächst habe ich einen Ansatz zur Erkennung informativer Nachrichten während katastrophaler Ereignisse mit Hilfe von Deep-Learning-Techniken vorgestellt. Durch die automatische Erkennung informativer Nachrichten, die von Nutzern bei Großereignissen gepostet werden, können Fachleute, die mit dem Krisenmanagement befasst sind, Schäden besser einschätzen, und sofort ausschließlich anhand der relevanten Informationen auf den Social-Media-Kanälen handeln. Außerdem habe ich eine Studie zur Analyse von Twitter-Nachrichten durchgeführt, die während der Covid-19-Pandemie gepostet wurden. Zunächst habe ich vier Millionen Tweets gesammelt, die seit Beginn der Pandemie in portugiesischer Sprache gepostet wurden, und eine Analyse der Debatte über die Pandemie erstellt. Ich verwandte Themenmodellierung, Stimmungsanalyse und Hashtag-Empfehlungstechniken um die Online-Diskussion über die Covid-19-Pandemie zu beleuchten.

**Schlagwörter:** Entity Linking, Ensemble Learning, Wissensbasis, Deep Learning

## ACKNOWLEDGMENTS

First and foremost, I would like to thank God, who has granted me countless blessings and opportunities, so that I have been able to accomplish this work. Besides my own efforts, the success of this thesis depends largely on the encouragement, support and guidelines of many other wonderful human beings whom I have crossed paths with.

I would like to express my greatest gratitude to my supervisor, Prof. Dr. techn. Wolfgang Nejdil for giving me the opportunity of being part of L3S Research Center and Gottfried Wilhelm Leibniz University of Hannover, for guiding me in how to pursue excellent research, and for supporting me during all these past years.

Throughout the course of my PhD studies, I have worked with many great researchers and colleagues who have supported me in various aspects. I have met wonderful people from other countries with different backgrounds and beautiful life stories.

I would like to thank my brother Rafael Stoffalette João, my father Antonio Carlos João and my mother Joana Aparecida Stoffalette João for their unconditional love and support in every aspects.

My work was partially supported by CNPq (Brazilian National Council for Scientific and Technological Development) under grant GDE No. 203268/2014-8 and the European Commission for the ERC Advanced Grant ALEXANDRIA under grant No. 339233. Without these grants such accomplishment would not be possible.

Last but not least, my beloved girlfriend Patricia Prado Munhoz, my partner in many important life decisions who has supported me and comforted me during difficult times.

## FOREWORD

During the course of my Ph.D studies, I published papers on knowledge representation, information retrieval, natural language processing and machine learning. The contributions presented in this thesis have been published in the following venues:

- The contributions in Chapter 2 on generating difficulty labels for an entity mention have been published in:
  - [JFD19] João, R.S., Fafalios, P. and Dietze, S., 2019, April. Same but different: distant supervision for predicting and understanding entity linking difficulty. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 1019-1026).
- The contributions in Chapter 3 on the prior probability of an entity mention computed over snapshots of Wikipedia at different points in time as well as an approach for temporal entity linking have been published in:
  - [SJ20] João, R.S., 2020, April. On the Temporality of Priors in Entity Linking. In European Conference on Information Retrieval (pp. 375-382). Springer, Cham.
  - [SJ17] João, R.S., 2017. Time-Aware Entity Linking. In DC@ International Semantic Web Conference.
- The contributions in Chapter 4 on an ensemble learning approach for improving precision by predicting the most correct entity linking system considering the particular characteristics of each particular mention has been published in:
  - [JFD20] João, R.S., Fafalios, P. and Dietze, S., 2020, March. Better together: an ensemble learner for combining the results of ready-made entity linking systems. In Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing (pp. 851-858).
- The contributions in Chapter 5 on a web-based search interface which enables non-expert users to interact with the archived documents without the need to knowing how to formulate complex queries using the SPARQL language have been published in:

- [SJ21b] João, R.S., 2021, March. A Semantic Layer Querying Tool. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (pp. 1101-1104).
- The contributions in Chapter 6 on the viability of machine learning approaches for developing an automatic mechanism to classify tweets according to their informativeness during catastrophic events have been published in:
  - [SJ21a] João, R.S., 2021. On Informative Tweet Identification for Tracking Mass Events. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2 (pp. 1266-1273).
- The contributions in Chapter 7 which deals with methods and approaches for an analytical perception of the Covid pandemic on Twitter is under submission





# Contents

<b>Table of Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Entity Linking . . . . .	1
1.1.1 Problem Formulation . . . . .	2
1.1.2 Named Entity Recognition . . . . .	4
1.1.3 Word Sense Disambiguation . . . . .	5
1.1.4 Wikification . . . . .	5
1.2 Contributions . . . . .	5
1.3 Thesis Structure . . . . .	6
<b>2 Predicting and Understanding Entity Linking Difficulty</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Motivation . . . . .	11
2.3 Learning Entity Linking Difficulty . . . . .	12
2.3.1 Problem Formulation . . . . .	12
2.3.2 Labeling Process . . . . .	12
2.4 Features . . . . .	13
2.4.1 Mention-based features . . . . .	14
2.4.2 Document-based features . . . . .	16

---

2.4.3	Temporal features . . . . .	16
2.5	Evaluation . . . . .	17
2.5.1	Corpus . . . . .	17
2.5.2	Labeling . . . . .	17
2.5.3	Quality of the generated labels . . . . .	18
2.5.4	Balancing & Sampling . . . . .	19
2.5.5	Classification Models . . . . .	19
2.5.6	Baselines and Multifeature Approach . . . . .	19
2.5.7	Configurations . . . . .	20
2.5.8	Evaluation Metrics . . . . .	20
2.6	Evaluation Results . . . . .	20
2.6.1	Classification Performance . . . . .	20
2.6.2	Influence of Dataset Size . . . . .	21
2.6.3	Feature Analysis . . . . .	23
2.7	Impact on Entity Linking . . . . .	25
2.8	A Lightweight System for Entity Disambiguation . . . . .	27
2.9	Conclusions and Future works . . . . .	28
<b>3</b>	<b>Temporality of Prior Probability in Entity Linking</b>	<b>31</b>
3.1	Introduction . . . . .	31
3.2	Problem Definition . . . . .	32
3.2.1	Candidate Entities Generation and Ranking . . . . .	33
3.3	Experiments . . . . .	34
3.3.1	Datasets . . . . .	34
3.3.2	Prior Probability Changes . . . . .	34
3.3.3	Comparing Ranked Entities . . . . .	35
3.3.4	Top Ranked Entity Changes . . . . .	36
3.3.5	Top 5 Entities Changes . . . . .	36
3.4	Conclusions . . . . .	37
<b>4</b>	<b>An Ensemble Learner for Combining Entity Linking Systems</b>	<b>39</b>
4.1	Introduction . . . . .	39
4.2	Problem definition . . . . .	41
4.3	MetaEL+ . . . . .	41
4.3.1	Features . . . . .	42
4.3.2	Surface Form-based Features . . . . .	42

---

4.3.3	Mention-based Features . . . . .	43
4.3.4	Document-based Features . . . . .	43
4.3.5	Classifiers . . . . .	43
4.3.6	Training and Labeling . . . . .	44
4.4	Evaluation . . . . .	44
4.4.1	Datasets . . . . .	44
4.4.2	Entity Linking Tools . . . . .	45
4.4.3	Baseline and MetaEL+ Methods . . . . .	45
4.4.4	Evaluating EL performance . . . . .	46
4.4.5	Evaluating the classification performance . . . . .	46
4.5	Results . . . . .	47
4.5.1	Annotation and agreement statistics . . . . .	47
4.5.2	Upper bound performance . . . . .	48
4.5.3	Entity Linking Performance . . . . .	49
4.5.4	Prediction Performance . . . . .	50
4.5.5	Binary classification . . . . .	51
4.5.6	Feature Analysis . . . . .	52
4.5.7	Synopsis and Limitations . . . . .	53
4.6	Related Works . . . . .	54
4.7	Conclusions . . . . .	55
<b>5</b>	<b>A Semantic Layer Querying Tool</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Related Works . . . . .	58
5.3	Open Web Archive Data Model . . . . .	59
5.4	System Architecture . . . . .	60
5.5	Interface Design . . . . .	61
5.6	Querying the semantic layer . . . . .	64
5.7	Conclusions and Future Works . . . . .	65
<b>6</b>	<b>Informative Tweet Identification</b>	<b>67</b>
6.1	Introduction . . . . .	67
6.2	Related Works . . . . .	69
6.3	Methodology . . . . .	69
6.3.1	Traditional models . . . . .	70
6.3.2	Features . . . . .	70

---

6.3.3	Text-based features . . . . .	70
6.4	User-based features . . . . .	71
6.4.1	Deep learning approaches . . . . .	72
6.4.2	Word embedding methods . . . . .	72
6.4.3	Text embedding methods . . . . .	73
6.4.4	BERT . . . . .	74
6.4.5	Dropout layers . . . . .	75
6.4.6	A Hybrid BERT model . . . . .	76
6.5	Evaluation Setup . . . . .	76
6.5.1	Datasets . . . . .	76
6.5.2	Evaluation Metrics . . . . .	78
6.5.3	Experiment settings . . . . .	78
6.6	Results . . . . .	78
6.7	Conclusions and Future Works . . . . .	81
<b>7</b>	<b>An Exploratory Analysis of Portuguese Tweets. Insights from Topics and Hashtags during Covid-19 Pandemic</b>	<b>83</b>
7.1	Introduction . . . . .	83
7.2	Related Works . . . . .	85
7.3	Methods . . . . .	85
7.3.1	Data Acquisition . . . . .	86
7.3.2	Preprocessing . . . . .	86
7.3.3	Text Representation . . . . .	87
7.3.4	Deep Transfer Learning . . . . .	88
7.4	Experiments and Analysis . . . . .	89
7.4.1	Corpus analysis . . . . .	89
7.4.2	Popular domains . . . . .	90
7.4.3	Words being used . . . . .	91
7.4.4	Topic Modeling . . . . .	92
7.4.5	Hashtags Suggestion . . . . .	94
7.4.6	Sentiment Analysis . . . . .	94
7.5	Conclusions . . . . .	98
<b>8</b>	<b>Conclusions and Future Works</b>	<b>99</b>
8.1	Conclusions . . . . .	99
8.2	Future Works . . . . .	100

**Bibliography**



## List of Figures

1.1	Snippet of text annotated with entities from Wikipedia. . . . .	2
2.1	Example of HARD entity mention (Kashmir). . . . .	14
2.2	Commonly recognized mentions in the New York Times corpus. . . . .	18
2.3	Influence of dataset size on prediction performance (macro average) using Random Forest. . . . .	23
2.4	Attribute importance (Mean Decrease Impurity) per feature for SAMPLE25. . . . .	24
2.5	Correlation among features (Pearson's r). . . . .	25
2.6	Effect of human feedback on the accuracy of semi-automated EL systems for different proportion of human judgments: 5% (left), 10% (middle), and 15% (right). . . . .	27
2.7	Landing page for text input and agreement calculation. . . . .	28
2.8	Example of ambiguous entity mention (Jaguar). . . . .	29
5.1	Semantic Layers querying system architecture. . . . .	60
5.2	Search page with query suggestion and advanced search. . . . .	62
5.3	Results page with a list of surrogates. . . . .	63
6.1	BERT's architecture. . . . .	75
6.2	BERT's hybrid model architecture. . . . .	76
7.1	Monthly distribution of tweets and retweets . . . . .	89
7.2	Most mentioned Twitter accounts . . . . .	90
7.3	Most influential user accounts . . . . .	90

7.4	Top 10 most popular domains . . . . .	92
7.5	Top 10 n-grams . . . . .	93
7.6	Word clouds for the topics <i>coronavirus</i> , <i>hidroxicloroquina</i> and <i>fiqueem-casa</i> respectively . . . . .	93



## List of Tables

2.1	Summary of features from different categories. . . . .	15
2.2	Overall prediction performance (macro average) using SAMPLE25. . .	21
2.3	Prediction performance per class using SAMPLE25 with unbalanced training. . . . .	22
2.4	Prediction performance per class using SAMPLE25 with balanced training. . . . .	22
3.1	Information about the Wikipedia editions used for mining mention and entities. #Pages refers only to the number of entities' pages, excluding special pages. . . . .	33
3.2	#Docs is the number of documents. Docs Year is the documents' publication time. #Annotations is the number of annotations (Number of non-NIL annotations). Annot. Year is the reference KB time period where the annotations were taken from. . . . .	34
3.3	Accuracy of the models on different datasets across different time periods. . . . .	35
3.4	A mention example and its top 5 ranked candidate entities captured from two Wikipedia editions. . . . .	37
4.1	Ground truth datasets main statistics. . . . .	45
4.2	Performance of the used EL tools on CONLL [HYB <sup>+</sup> 11] . . . . .	45
4.3	Annotation statistics of the test datasets. . . . .	47
4.4	Entity linking performance. . . . .	49
4.5	Performance of multi-label classification . . . . .	51
4.6	Performance of binary classification. . . . .	52

---

4.7	Effectiveness of different feature combination using METAEL+LOOSE on CONLL. . . . .	53
6.1	Examples of tweets from CRISIMMD[AOI18] dataset. . . . .	68
6.2	Complete datasets classes distributions. . . . .	77
6.3	Subsets classes distribution. . . . .	77
6.4	Models performance on the original datasets. . . . .	80
6.5	Models performance on the subsets. . . . .	82
7.1	List of seed hashtags . . . . .	87
7.2	Month-wise tweets distribution . . . . .	91
7.3	Month-wise hashtags distribution . . . . .	91
7.4	Most similar hashtags to the hashtag <i>#vacina</i> . . . . .	95
7.5	Macro Acc,P,R,F . . . . .	97
7.6	BERT fine tuned model per class prediction . . . . .	97

In this chapter I briefly introduce the entity linking task as well as its variants followed by a formal definition and an overview of the main components of an entity linking pipeline.

## 1.1 Entity Linking

Natural language processing (NLP) refers to the branch of computer science which is concerned with the ability to make computers to understand human language in both written and spoken forms as similar as possible as humans do. The human language is full of ambiguities that makes it difficult to write computer programs that can accurately interpret the intended meaning of written or spoken texts. Therefore NLP methods try to convert unstructured language data into a structured format and enable computers to understand texts and originate relevant information.

There are innumerable tasks involving NLP which try to make sense of human language and understand the contexts in which words are used: Named entity recognition (NER) [NS07, YB19, LSHL20], Entity Linking (EL) [SWH14, SSA<sup>+</sup>20], Word Sense Disambiguation (WSD) [Nav09], Co-reference resolution [Ela05], Relationship Extraction (RE) [BB07], Part-of-speech tagging (PoS tagging) [Mar12], Sentiment analysis [MHK14, ZWL18], Text classification [AZ12, KJMH<sup>+</sup>19], Question-Answering (QA) [KM11, BDDM15], among others.

One of the main topics in this thesis is concerned with the entity linking task, which is the task of recognizing entity mentions in texts and linking them to the corresponding entries in a reference knowledge repository (KB). The output of this process is illustrated in Figure 1.1. Here the term "Ayrton Senna da Silva" should be linked to the famous brazilian racing driver, the term "Formula One" should be linked to the class of single-seater auto racing and "Brazil" to the largest country in South America.



Figure 1.1: Snippet of text annotated with entities from Wikipedia.

While the EL process may seem relatively easy for humans, it poses several difficulties for machines to recognize and fully understand the real meaning of an entity. Take for instance the mention "Kashmir" appearing in one text document. It can refer to any of the entities "Kashmir" (1975 song by rock band Led Zeppelin), "Kashmir" (the northwestern region of the Indian territory), "Kashmir" (a type of wool made from cashmere and pashmina goats), among many others. Based on the document's content, one of these candidate entities is selected and linked to the corresponding entry in the knowledge repository.

### 1.1.1 Problem Formulation

The entity linking task is formalized as follows. Consider a document  $d$  from a set of documents  $D = \{d_1, d_2, \dots, d_n\}$ , and a set of mentions  $M = \{m_1, m_2, \dots, m_n\}$  extracted from  $d$ . The goal of the entity linking task is to find a unique identity represented by an entity  $e$  from a set of entities  $E = \{e_1, e_2, \dots, e_n\}$ , with relation to each mention  $m$ . The set of entities  $E$  is usually extracted from a reference knowledge base ( $KB$ ) or a catalog of entities.

DBPedia [LIJ<sup>+</sup>15], YAGO [SKW07], Freebase [BEP<sup>+</sup>08], WordNet [F<sup>+</sup>98] and Wikidata [VK14], are just a few examples of reference knowledge bases that have grounded many entity linking tasks due to the fact they contain a rich set of information about entities.

A typical entity linking pipeline consists of the following three steps [HRN<sup>+</sup>13]: mention detection, candidate generation and disambiguation. Below we describe each

of these components in more details.

### Mention detection

One of the first steps in an entity linking pipeline is the so-called mention detection, also known as mention extraction, which is responsible for identifying snippets of text that can potentially link to entities in a reference *KB*. The task of mention detection is very related to the task of named entity recognition (NER), which by itself is another wide field of research(cf. Section 1.1.2) and can be performed with the assistance of NER techniques.

Formally, for an input document  $d$ , the set of mentions  $M_d$  is to be extracted, where each mention  $m \in M_d$  is delimited by its initial and ending character offsets. The majority of the entity linking systems rely on a dictionary of known surface forms to detect mentions [BP06, Cuc07, FS10, HYB<sup>+</sup>11, KSRC09, RRDA11, MJGSB11].

### Candidate Generation

The goal of this step is given an ambiguous entity mention  $m$ , to provide a list of its possible candidate entities. Formally, given a mention  $m$ , a candidate generation provides a list of possible entites  $e_1, e_2, \dots, e_n$  for each entity mention in a document. There are inumerous methods for candidate entity generation. Basically the most widely used approaches rely on surface form matching, expansions with aliases, and prior probability computation. In the first approach a list of candidate entities is composed of entities that match various surface forms of mentions in the text [LT19, MBB<sup>+</sup>17]. In the second approach a dictionary of additional aliases using metadata like the redirect pages from Wikipedia is used [FCL<sup>+</sup>19, ZSG16]. And the third approach for candidate generation is based on the prior probabilities, i.e. given a mention  $m$ , determine the prior probability of an entity  $e$  being the linking target computed on the Wikipedia entity hyperlinks [HYB<sup>+</sup>11].

It is also common to combine multiple approaches to the candidate generation step. Ganea and Hofmann [GH17] for instance proposed an approach that takes into account the prior probabilities calculated from the entities hyperlink statistics of CrossWikis [SC12] and Wikipedia as well as on entity aliases from relationships of the YAGO [HSB<sup>+</sup>11] ontology.

### Disambiguation

The disambiguation step is the last step in the pipeline of the entity linking process and consists of selecting a single entity from a list of candidate entites. Simply put, the disambiguation process can be seen as a ranking problem in which the top ranked candidate is selected from a list of candidate entities for every given mention  $m$  appearing in the input text.

Many approaches have been proposed to solve the ranking of candidate entities. Bunescu and Paşca [BP06] for instance employed a support vector machine model to rank the candidate entities, Varma et al. [VPK<sup>+</sup>09] proposed a disambiguator that ranked candidates based on the textual cosine similarity between the paragraph surrounding the mention and the text of the candidate page. Cucerzan [Cuc07] disambiguated the mention by taking the scalar product of the candidate vector and the extended document vector. Han and Zhao [HZ09] ranked the candidates based on the Bag-of-Words technique and the Wikipedia semantic knowledge similarity.

Basically we can distinguish between two main disambiguation strategies, the local approach in which each mention is disambiguated independently of the others [BP06, MC07], and the global approach where all mentions are disambiguated jointly [KSRC09, FS10, RRDA11, HYB<sup>+</sup>11]. The global disambiguation approach defines a coherence function across multiple entities in a context and attempts to solve the disambiguation problem collectively, however when it is cast as a graph problem, it becomes a NP-hard problem where approximations are required.

Moreover with the recent advances in the field of neural networks, modern disambiguation approaches have established the state-of-the-art, outperforming the engineered features based models. He et al. [HLL<sup>+</sup>13], Sun et al. [SLT<sup>+</sup>15], Yamada et al. [YSTT16], Ganea and Hofmann [GH17], Le and Titov [LT18], Yang et al. [YIR18], Radhakrishnan et al. [RTV18] are examples of published works that employ neural networks on the EL task.

Sevgili et al. [SSA<sup>+</sup>20] published a solid literature review of EL systems based on neural models and how they benefited from what the authors call the “*deep learning revolution*” in NLP.

### 1.1.2 Named Entity Recognition

Named entity recognition (NER), which is sometimes also called as entity extraction or entity identification is concerned with the process of detecting a word or a phrase that references a particular entity in texts. NER is an essential step in most NLP tasks. It first appeared in the Sixth Message Understanding Conference (MUC-6) [MUC95] as a subtask where its main goal was to identify proper names, acronyms and miscellaneous other identifiers that could be categorized as one of the three ENAMEX types: PERSON (named person or family), ORGANIZATION (named corporate, governmental, or other organizational entity) and LOCATION (name of politically or geographically defined location cities, provinces, countries, international regions, bodies of water, mountains, etc). Listing 1.1 shows an example of annotated text with ENAMEX entity types from MUC-6.

```
1 Mr.<ENAMEX TYPE="PERSON">Dooner</ENAMEX> met with <ENAMEX TYPE="
  PERSON">Martin Puris</ENAMEX>, president and chief executive
  officer of <ENAMEX TYPE="ORGANIZATION">Ammirati & Puris</ENAMEX>.
```

Listing 1.1: Annotated text with ENAMEX entity types.

### 1.1.3 Word Sense Disambiguation

The word sense disambiguation (WSD) is a very similar task to the entity linking task, however while in the EL task a mention can be linked to an entity that may or may not exist in a reference knowledge base, the word sense disambiguation task assumes there is a perfect match between mentions and word senses in a dictionary [Nav09]. Word sense disambiguation methods can be basically divided into three categories: supervised methods, unsupervised methods and knowledge base methods. The supervised methods use features extracted from manually sense-annotated corpus to train a supervised machine learning model. Unsupervised methods do not need sense-annotated text and usually rely on clustering algorithms. The knowledge based methods rely on external knowledge resources such as WordNet [F<sup>+</sup>98].

### 1.1.4 Wikification

Wikipedia is a very popular and trusted source of information in encyclopedia-like format. Wikipedia is a web-based project supported by the Wikimedia Foundation that is edited in a collaborative fashion by a community of anonymous users<sup>1</sup>.

Wikipedia has grounded many works in NLP and offered a new way to approach the problem of entity ambiguity. The term Wikification has been firstly used by Mihalcea and Csomai [MC07], but instead of clustering entities as is done in Coreference Resolution, mentions could be linked to encyclopedia pages. Therefore the Wikification task can be basically defined as the automatic annotation of text fragments (phrases) by linking them to their appropriate Wikipedia articles. The difference between the EL task and Wikification lies on the fact that while the EL task annotates real world entities, the Wikification task annotates entities as well as concepts as long as there are encyclopedia articles identifying them.

In order to encourage research on the wikification task, the INEX workshops ran a "Link the Wiki" task between 2007 and 2009 [HGT09].

## 1.2 Contributions

In the first part of this thesis I focus basically on the semantic enrichment of text documents. The contributions appearing in Chapter 2 proposes a distant supervision method for predicting and understanding the entity linking difficulty. I propose an approximation of the difficulty to link a particular entity mention and pose it as a supervised classification task that is capable of predicting the EL difficulty of entity mentions using a variety of features.

An important component in entity linking approaches is the mention-to-entity prior probability. Even though there is a large number of works in entity linking, the

---

<sup>1</sup><https://en.wikipedia.org/wiki/Wikipedia:About>

existing approaches do not explicitly consider the time aspect, specifically the temporality of an entity’s prior probability. In Chapter 3 I posit that this prior probability is temporal in nature and affects the performance of entity linking systems. I perform a systematic study on the effect of the prior on the entity linking performance over the temporal validity of both texts and reference knowledge bases.

The contributions in Chapter 4 is based on the assumption that the performance of the entity linking process may be optimised by exploiting results from distinct EL systems on the same corpus, thereby leveraging their individual strengths on a per-mention basis. To this end I propose an ensemble learning approach for improving precision by predicting the most correct entity linking system considering the particular characteristics of each particular mention.

In Chapter 5 I implemented a web-based search interface which enables non-expert users to interact with archived documents without the need to knowing how to formulate complex queries using the SPARQL language.

In the second part of this thesis I explore methods and propose solutions to analyse social media messages during crisis events. In Chapter 6 I study the viability of machine learning approaches for developing an automatic mechanism to classify tweets according to their informativeness during catastrophic events. Moreover I propose a hybrid model, namely BERTHyb, that combines both handcrafted features with the ones learned by deep learning method. I demonstrate that the proposed solution is more effective in identifying informative Twitter messages than conventional classifiers in different crisis related corpus. The contributions in Chapter 7 deals with methods and approaches for an analytical perception of the Covid-19 pandemic on Twitter. I investigate the online debate taking place on social media with respect to Covid-19 in Brazil. I describe a Portuguese Twitter dataset with more than 4 million tweets collected during a 16-month period and search for insights with descriptive textual analytics and data visualization approaches such as the wordclouds.

## 1.3 Thesis Structure

The remainder of this thesis is organized as follows.

In Chapter 2 I introduce the concept of difficulty of a mention to be linked and I present an automated approach to generate difficulty labels relying on the agreement among different entity linking systems. I introduce a set of features, i.e. mention-based, document-based and temporal-based features that detect latent characteristics that affect entity linking performance. I demonstrate that entity linking difficulty can be estimated on the fly with high precision ( $>0.83$ ) and recall ( $>0.72$ ) even using a limited amount of the available training data.

In Chapter 3 I provide a study of the prior probability on the entity disambiguation problem computed over snapshots of Wikipedia at different points in time. I show that the priors change over time and the overall disambiguation performance using



temporal priors show high variability, bringing us to the conclusion that the temporal effects should be not only taken into account in (a) building entity linking approaches, but have major implications in (b) evaluation design, when baselines that are trained on temporally distant knowledge sources are compared.

In Chapter 4 I propose a novel approach where outputs of multiple end-to-end entity linking systems are combined using an ensemble learning method for providing an improved set of entity links for a given corpus. I apply the proposed approach to three established datasets and demonstrate significant performance improvements compared to both the individual systems and six baseline strategies.

In Chapter 5 I develop and implement a web-based search interface that enables non-expert users to interact with the archived documents without the need to knowing how to formulate complex queries using the SPARQL language to answer basic information needs. The main goal in this work is to assist users in the expression of their information needs by simply typing free text keywords and retrieving documents from archived collections.

In Chapter 6 I proposed a hybrid model that combines a BERT-based model with handcrafted features for the problem of identifying informative tweets during catastrophic events.

In Chapter 7 I look at the conversation taking place on social media, specifically Twitter, with respect to Covid-19. I describe a Portuguese Twitter dataset with more than 4 million tweets collected during a 16-month period. I search for insights with descriptive textual analytics and data visualization, such as exploratory Word Clouds. I investigate popular keywords shared among Twitter users and I also apply topic modeling and sentiment analysis methods to investigate questions related to the topics evolution over time as well as the sentiment expressed by users during the pandemic.

Finally, in Chapter 8, I provide some conclusions to the thesis and enumerate some of the contributions as well as discuss some possible research directions associated with the topics presented in this thesis.



## Predicting and Understanding Entity Linking Difficulty

In this chapter I introduce an automated method towards generating difficulty labels for entity mentions in arbitrary corpora. The difficulty labels are based on the assumption that EL difficulty varies per corpus and also with each individual mention.

### 2.1 Introduction

Entity linking (EL), or named entity recognition and disambiguation (NERD), is the task of determining the identity of entity mentions in texts, thereby linking a mention to an entity within a reference Knowledge Base (KB), such as Wikipedia. EL is a crucial task of relevance for a wide variety of applications, such as Web search, information retrieval, or document classification. Usually, high precision (P) and recall (R) is required if EL results are to have a positive impact on any such application.

However, EL remains a challenging task. Systems differ along multiple dimensions and are evaluated over different datasets [SWH14], while their performance differs significantly across domains and corpora [RUNN17]. EL difficulty varies per corpus but also with each individual mention, where previous work has shown that mentions which are difficult to link often share common characteristics [HSN<sup>+</sup>12]. Typical examples include highly ambiguous mentions where a large number of potential candidates exists, mentions of long-tail entities which are not well represented in KBs, such as local public figures, or mentions whose meaning changes over time.

Given that automated EL pipelines never reach perfect P/R on arbitrary corpora, human judgments are often required to improve automatically generated EL results [CXQ15, SHA16, DDC12]. Therefore, estimating a priori the difficulty of linking a particular mention can facilitate high P/R systems, e.g. by flagging critical mentions which require manual judgments as part of semi-automated EL approaches. Such

approaches utilise the scalability of automated linkers wherever possible and benefit from the precision of human judgments to handle challenging cases. In this context, in particular the widely used practice of applying state-of-the-art EL systems out of the box calls for methods that enable detecting difficult to link mentions as well as latent characteristics that affect the EL performance, thus addressing the strong context-specific nature of EL.

In this chapter, we first introduce an automated method to generate difficulty labels (*HARD*, *MEDIUM*, *EASY*) for entity mentions in an arbitrary corpus. The proposed method utilises agreement and disagreement measures obtained by applying state-of-the-art EL systems on the given corpus. Experimental results demonstrate the effectiveness of this labeling strategy on improving the performance of semi-automated EL, by enabling the efficient prediction of critical cases which require manual labeling (e.g., from domain experts or through crowdsourcing).

To detect characteristics that determine the difficulty of a mention to be linked correctly, as well as to allow predicting EL difficulty on-the-fly (e.g., for cases where real-time analysis is needed, or when no labels can be assigned using the proposed labeling method), we exploit the generated difficulty labels as training data for a multi-class classification task capable of predicting the EL difficulty of entity mentions using a diverse feature set. Through an extensive feature analysis we investigate the importance of different types of features, inspired by previous work as well as by the observed characteristics of difficult-to-link mentions.

We apply our approach to the New York Times (NYT) corpus [San08] and find that the position of the mention in the document, the size of the sentence containing the mention, and the frequency of the mention in the document (all related to the mention’s context) are the three most useful features for predicting EL difficulty in our experiments, while temporal features also contribute. In addition, we demonstrate that EL difficulty can be estimated on the fly with high precision ( $>0.83$ ) and recall ( $>0.72$ ) even using a limited amount of the available training data (below 25% of the original data), while recall can be further improved using a balanced training dataset. While to the best of our knowledge no works exist which address this prediction task, we compare our configurations to two baselines which utilise few but highly predictive features (number of candidate entities, mention length) and show superior performance of our multi-feature approach. In a nutshell, we make the following contributions:

- We introduce an automated approach to generate difficulty labels which relies on agreement information among different EL systems. The generated labels can be used to improve semi-automated EL, as direct indicators or through distant supervision.
- We propose a novel approach, feature sets and classifiers for predicting EL difficulty as well as for detecting latent, corpus-specific characteristics that affect EL performance.

The rest of the chapter is organised as follows: Section 2.2 motivates the problem and discusses related works. Section 2.3 introduces the proposed method to assign difficulty labels. Section 4.3 describes the features used in our multi-class classification task. Section 2.5 reports experimental results on predicting and understanding EL difficulty. Section 2.7 shows how the proposed method can improve semi-automated EL. Finally, Section 2.9 concludes the chapter and discusses interesting directions for future research.

## 2.2 Motivation

Whereas both users and applications of automatically generated entity annotations usually require high performance, in particular, high precision, EL remains a challenging task, where no single system has yet emerged as de-facto-standard. Evaluations using the GERBIL benchmark [RUNN17], a framework that compares EL systems over a large number of ground truth datasets, have shown that their performance is highly affected by the characteristics of the datasets, like the number of entities per document, the document length, the total number of entities, or the salient entity types [URN15]. This demonstrates that, the widely used practice of applying state-of-the-art EL systems out of the box, i.e. without corpus-specific training, usually does not provide the best performance.

In particular, wrongly linked mentions often share certain common characteristics, where typical examples include: i) highly ambiguous mentions which often have a large number of candidate entities and/or are short (e.g. family names “Brown” or “Williams”); ii) mentions of long-tail entities (often not represented in reference KBs, e.g. regional politicians); iii) mentions of entities where the respective meaning evolves significantly over time (e.g. “Germany” before or after 1990, or “President of the US”); iv) mentions of entities where the popularity, and hence prior probability, of disambiguation candidates changes significantly over time (like “Amazon” in 1980 or 2018); v) mentions which are prone to partial matching, such as location names (e.g. “Madrid” which may refer to the city or the football club Real Madrid depending on the context).

These features underline the corpus-specific nature of EL difficulty. For these reasons, when applying any state-of-the-art system to an arbitrary corpus, estimating the actual quality of the produced annotations remains challenging. In addition, independent of the overall performance, real-world applications which utilise annotations call for quality standards which cannot necessarily be met by automated EL approaches alone. Thus, estimating a priori the difficulty of linking a particular mention can facilitate high precision systems, e.g. by flagging critical mentions which require manual judgments as part of semi-automated EL approaches [CXQ15, SHA16, DDC12].

## 2.3 Learning Entity Linking Difficulty

We firstly define the problem of entity linking difficulty learning followed by the description of the labeling process.

### 2.3.1 Problem Formulation

Let  $D$  be a corpus of documents, e.g., a set of news articles, covering the time period  $T_D$ . Consider also a contemporary KB  $K$ , for instance Wikipedia, describing information for a set of entities  $E$ . The output of applying EL on the documents of  $D$  is a set of annotations of the form  $\langle d, m, p, e \rangle$ , where  $d$  is a document in  $D$ ,  $m$  is an entity mention in  $d$  (a word or a sequence of words),  $p$  is the position of  $m$  in  $d$ , and  $e$  is an entity in  $E$  that determines the identity of  $m$ .

We now define the problem of determining the difficulty in linking a mention  $m$  to an entity in  $K$  as a multi-class classification problem where  $m$  is assigned to one of the following classes:

- *HARD*: Difficult to disambiguate mention (state of the art EL systems usually fail to find the correct link)
- *EASY*: Easy to disambiguate mentions (state of the art EL systems almost always find the correct link)
- *MEDIUM*: All other cases (neither EASY nor HARD)

Below we describe an automated approximation strategy to assign these difficulty labels on entity mentions of an arbitrary corpus.

### 2.3.2 Labeling Process

We propose to use freely available state-of-the-art EL systems  $\langle s_1, \dots, s_n \rangle$  which operate on the same reference KB  $K$  (e.g., Wikipedia 2016) and are applied to the same corpus  $D$ . The degree of agreement of all systems  $s_i$  is then used as indicator of the EL difficulty.

In particular, assuming  $n = 3$  systems, three sets of entity links are produced ( $A_1$ ,  $A_2$ , and  $A_3$ ). To generate the labels we consider only the commonly recognised entities, i.e., the mentions for which all three systems provide a link, which may or may not be the same. The set  $A$  of common entity annotations has elements of the form  $\langle d, m, p, e_1, e_2, e_3 \rangle$  where  $d$  is the document,  $m$  is the mention,  $p$  is the position of  $m$  in  $d$ , and  $e_1$ ,  $e_2$ , and  $e_3$  are the entities provided by  $s_1$ ,  $s_2$ , and  $s_3$ , respectively.

A mention  $m_i$  is assigned with the *HARD* label if all three systems disagree, i.e. each one provides a link to a different entity  $e_j$ . The intuition is that in this case,

at least 2/3 systems failed to find the correct entity. Formally, for  $n = 3$  the set of HARD annotations  $A^H$  is defined as:

$$A^H = \{\langle d, m, p, e_1, e_2, e_3 \rangle \in A \mid e_1 \neq e_2 \neq e_3\} \quad (2.1)$$

As *EASY* we consider the cases where all systems agree on the same mention, i.e., all provide the same entity link. Formally:

$$A^E = \{\langle d, m, p, e_1, e_2, e_3 \rangle \in A \mid e_1 = e_2 = e_3\} \quad (2.2)$$

As *MEDIUM* we consider all other cases:

$$A^M = \{a \in A \mid a \notin A^H \wedge a \notin A^E\} \quad (2.3)$$

i.e., cases where exactly 2/3 systems provide the same entity.

In Figure 2.1 we show one example of ambiguous mention in which the the 3 EL systems disagreed on the correct entity and thus it is considered as a HARD mention in this context.

It is obvious that the above labeling process can provide wrong approximations since it assumes that if the systems provide the same entity link then this link is correct. Our assumption is that, in particular the EASY class might contain false positives to a certain degree, e.g. when all systems agree on the same but wrong entity. In Section 2.5.2 we provide evaluation results of the quality of class assignments obtained through our approach, suggesting a precision of more than 93% on average given our experimental setup.

An additional limitation arises from the fact that this labeling method requires mentions to be recognised by all the considered systems, i.e., it cannot provide labels for mentions recognised by only one or two of the systems. As shown in our experiments (Sect. 2.5.2), the common mentions are less than 30% of the total mentions recognised by each system, thus we need to predict the EL difficulty of all other mentions. Furthermore, the efficiency of this labeling method depends on the efficiency of the used systems, thus it might not be applicable for cases where real-time analysis is needed or large amounts of documents are to be annotated.

## 2.4 Features

To address the issues of the aforementioned labeling strategy, supervised classification can be used to predict EL difficulty. In particular, a distantly supervised classification model may be trained using the proposed labeling strategy in order to learn to predict the linking difficulty of arbitrary entity mentions. For this, we need a diverse set of features which covers different aspects of EL difficulty.

Inspired by previous works as well as by the observed characteristics of difficult to link mentions which are not correctly disambiguated through state-of-the-art systems

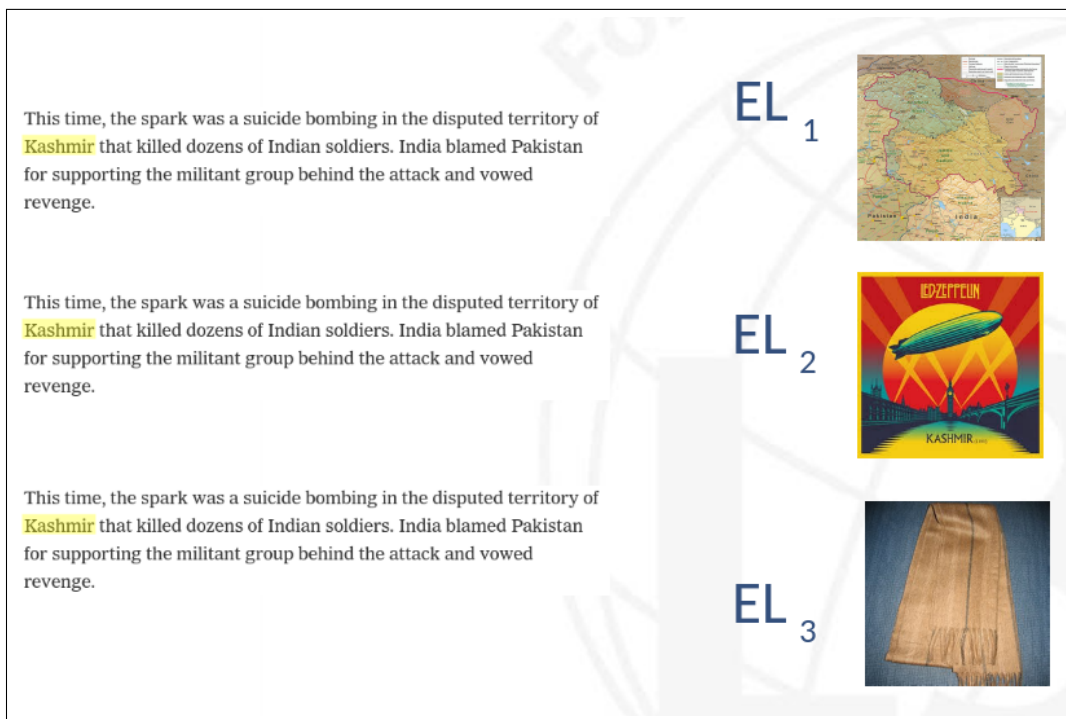


Figure 2.1: Example of HARD entity mention (Kashmir).

(cf. Section 2.2), we consider features of the following three categories: i) *mention-based* (features of the mention itself), ii) *document-based* (features of the document containing the mention), and iii) *temporal* (features that consider the temporal evolution of either the mention or the document containing the mention). Below we detail each of them, while a summary is given in Table 2.1.

### 2.4.1 Mention-based features

**Mention length** ( $m_{len}$ ): the number of mention’s characters. Short mentions are usually more ambiguous compared to long mentions (e.g., *Adams* vs *Schwarzenegger*).

**Mention words** ( $m_{words}$ ): the mention’s number of words. Unigram mentions are usually more ambiguous than mentions with more than one word (e.g., *John* vs *John McCain*).

**Mention frequency** ( $m_{freq}$ ): the number of mention occurrences within the document. More occurrences imply that the document is closely related to the mention, thus the context of the mention is more likely to be related to the actual mention.

**Mention document frequency** ( $m_{df}$ ): the number of documents in the corpus  $D$  containing at least one occurrence of the mention. Higher  $m_{df}$  implies popularity of the term(s), suggesting that more context is available about this mention.

**Mention candidate entities** ( $m_{cand}$ ): the number of candidate entities in the



Table 2.1: Summary of features from different categories.

Category	Notation	Description
Mention	$m_{len}$	Num of mention’s characters (length).
	$m_{words}$	Num of mention’s words.
	$m_{freq}$	Num of mention’s occurrences in the doc (frequency).
	$m_{df}$	Num of docs in the corpus containing at least one occurrence of the mention.
	$m_{cand}$	Num of mention’s candidate entities in a reference KB.
	$m_{pos}$	Mention’s normalised position in the doc (num of chars from the start of the doc / total num of doc’s chars).
Document	$m_{sent}$	Num of chars of the sentence containing the mention.
	$d_{words}$	Num of words in the document containing the mention.
	$d_{topic}$	Main topic discussed in the document containing the mention (e.g., SPORTS, or POLITICS).
Temporal	$d_{ents}$	Num of entity mentions recognised in the document containing the mention.
	$t_{age}$	The distance (age) of the doc’s publication date from the date of the reference KB.
	$t_{df}$	Number of docs containing at least one occurrence of the mention, published within $k$ intervals from the doc’s publication date (e.g., $+/- 6$ months).
	$t_{j_{min}} / t_{j_{max}} / t_{j_{avg}}$	Min, max and avg Jaccard similarity of the mention’s top-K similar words (computed using Word2Vec) for all pairs of consecutive time periods of fixed granularity.

reference KB. The articles in Wikipedia (the most common reference KB) contain hyperlinks with anchor texts pointing to entities, making it an important source for mining mention and entity relations. For a mention  $m$  we select as candidate entities those that appear as link destinations for  $m$ . A higher number of candidate entities indicates a more ambiguous mention.

**Mention’s normalised position ( $m_{pos}$ ):** the mention’s normalised position in the document, computed as the number of characters from the start of the document divided by the total number of document’s characters. Entities that appear early in the document are usually salient and representative for the document, indicating more representative context to facilitate their disambiguation.

**Mention’s sentence size ( $m_{sent}$ ):** The number of characters of the sentence containing the mention, specifically the length of the text between two punctuation marks containing the mention (considering only the punctuation marks “.”, “!”, “?”, “;”). An EL system may exploit the sentence containing the mention for disambiguating the entity, where larger sentences indicate more representative context for a particular mention.

### 2.4.2 Document-based features

**Document size** ( $d_{words}$ ): the number of words of the document containing the mention. Small documents do not provide much context information what hinders precise disambiguation of its entity mentions.

**Document topic** ( $d_{topic}$ ): the main topic (subject) discussed in the document containing the mention, selected from a predefined list of topics (like SPORTS, POLITICS, etc.). This information can be obtained either through an automated document classification algorithm or directly through the document’s metadata (if such information is available). The difficulty to disambiguate mentions varies among topics, for instance, related to the specificity of the topic or the prevalence of long-tail entities.

**Document’s recognised entities** ( $d_{ents}$ ): the total number of entities recognised in the document containing the mention. State of the art EL systems jointly disambiguate the entities in a document, e.g. by considering the linking structure in a reference KB. Thus, more recognised entities provide more contextual information enabling more precise disambiguation.

### 2.4.3 Temporal features

**Document publication age** ( $t_{age}$ ): the distance of the document’s publication date from the date of the reference KB (measured based on a fixed time interval, e.g., years). For example, if Wikipedia 2016 is the reference KB, a document of 2000 has age 16 while a document of 1990 has age 26. Mentions in old documents are more difficult to disambiguate since temporally distant entities are less well-represented or their context may have changed (e.g., linking the mention *Ronaldo* in a today’s article vs in an article of 1990’s).

**Mention’s temporal document frequency** ( $t_{df}$ ): the number of documents containing the mention, published within  $k$  intervals from the publication date of the document (e.g.,  $+/-$  6 months). Higher  $t_{df}$  means that the corresponding entity was popular during that particular time period, indicating the context of the mention is more likely to refer to the respective mention.

**Mention’s semantics stability** ( $t_{jmin}, t_{jmax}, t_{javg}$ ): the minimum, maximum, and average Jaccard similarity coefficient of the mention’s top-K similar words for all pairs of consecutive time intervals. The documents are grouped into a sequence of  $n$  time interval-specific subsets based on a fixed time granularity  $\Delta$  (e.g., year) and a Word2Vec Skipgram model [MCCD13] is trained for each group of documents (resulting in  $n$  different models). Given a mention, we retrieve its top-K similar words in each interval using the Word2Vec models and compute the Jaccard similarity of these sets of words for all pairs of consecutive time periods. We consider the minimum, maximum and average Jaccard similarity among all pairs. These three features consider the semantic evolution of terms, where the meaning of a term may change over time or the prior probability of a mention-entity link significantly changes

due to temporal events (e.g., *Germany* is likely to refer to Germany’s national football team during international football tournaments).

## 2.5 Evaluation

We evaluate the performance of supervised classification models on learning EL difficulty in a given corpus. The models make use of the proposed labeling strategy (cf. Section 2.3) and feature set (cf. Section 4.3) for i) predicting the EL difficulty of entity mentions, and ii) detecting corpus characteristics that affect the EL performance.

### 2.5.1 Corpus

We used the New York Times (NYT) Annotated Corpus [San08] which contains over 1.8 million articles published by the NYT between 1987 and 2007, covering a wide range of topics (like sports, politics, arts, business) and diverse content formats (like long texts, short notices, corrections, and headlines). The number of articles per year ranges from 79,077 (in 2007) to 106,104 (in 1987).

### 2.5.2 Labeling

We implemented the proposed labeling strategy (cf. Section 2.3.2) using the EL systems *Ambiverse* (previously AIDA) [HYB<sup>+</sup>11], *Babelfy* [MRN14], and *TagMe* [FS10]. In all three systems, we used Wikipedia 2016 as the common reference KB. For *Ambiverse*, we used its public Web API with the default configuration. For *Babelfy*, we used a local deployment and a configuration suggested by the *Babelfy* developers<sup>1</sup>. For *TagMe* we used a local deployment with the default configuration and a confidence threshold of 0.2 to filter out low quality annotations. We examined the performance of each system on the widely-used CoNLL-TestB ground truth [HYB<sup>+</sup>11]. *Ambiverse* achieved 81% precision and 65% recall, *Babelfy* 81% precision and 68% recall, and *TagMe* 79% precision and 53% recall. The performance of the systems is very close to the one reported in the literature for the same dataset.

The number of commonly recognised mentions among the three systems is 11,876,437, which corresponds to 30%, 11% and 21% of the total mentions recognised by *Ambiverse*, *Babelfy* and *TagMe*, respectively. Figure 2.2 shows a graph with the number of recognised mentions by the three entity linking systems as well as the number of commonly recognized mentions by each of the systems combination. We see that our labeling strategy cannot assign labels to a large number of mentions which have not been recognised by all three systems, thus we need to predict the linking difficulty of these mentions. From the common mentions, 340,238 (2.9%) are HARD (all systems

---

<sup>1</sup>The configuration is available at: <https://goo.gl/NHXVVQ>

disagree with each other), 9,070,517 (78.6%) are EASY (all systems provide the same entity) and 2,465,682 (21.4%) are MEDIUM (2/3 systems provide the same entity). We notice that the labels are highly unbalanced: the number of HARD cases is much smaller than the number of EASY and MEDIUM cases.

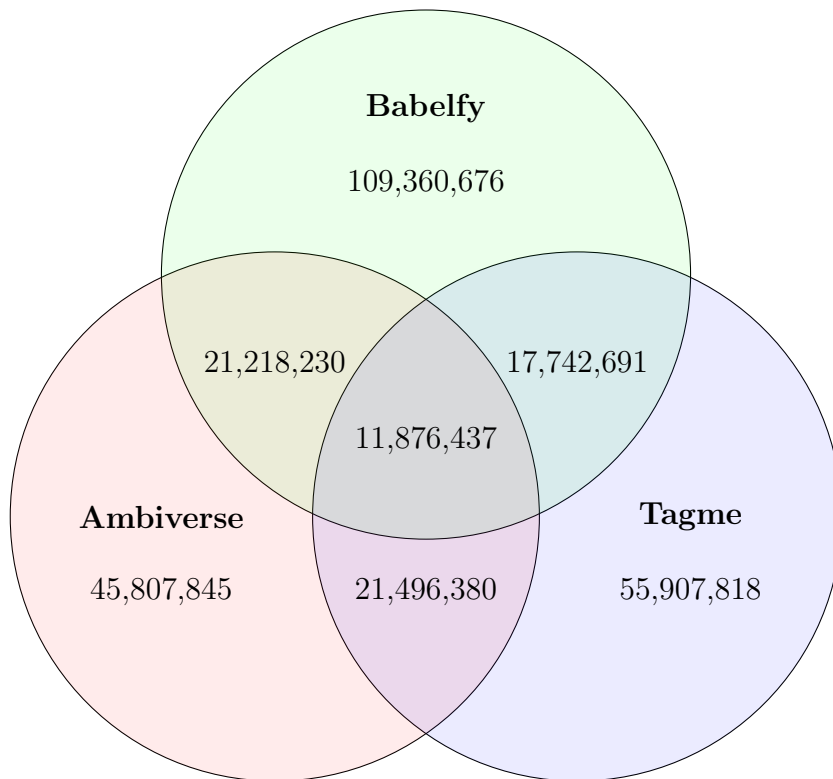


Figure 2.2: Commonly recognized mentions in the New York Times corpus.

### 2.5.3 Quality of the generated labels

First, we examined if the HARD mentions are indeed hard for all three systems or if there is one showing consistently high performance on these cases. We manually produced the ground truth for a random sample of 500 HARD cases. Ambiverse, Babelify and TagMe managed to find the correct entity in 24%, 16% and 31% of the cases, respectively. We notice that the joint effectiveness of all systems is low, supporting our labelling strategy. Then we examined the precision of the EASY and MEDIUM labels. We randomly selected 200 mentions from the EASY class and for each one we manually examined if the entity provided by the three systems is correct. The accuracy for this subset is 95%, i.e. only 5% of the mentions have been wrongly classified as EASY. Regarding the MEDIUM class, we randomly selected 200 mentions and tested if the two systems that agree provide the correct entity (if not, then these mentions can be considered HARD). In this case we found that 12% of the mentions

have been wrongly classified as MEDIUM. Considering that the majority (78.6%) of the not-HARD cases are EASY (following the original unbalanced distribution), we expect an error rate of MEDIUM and EASY labels of less than 7%.

The generated annotations as well as the ground truths of the aforementioned qualitative evaluation are made publicly available.<sup>2</sup>

### 2.5.4 Balancing & Sampling

To cater for the highly uneven class distribution, we experimented with both *unbalanced* and *balanced* training data. The unbalanced training dataset maintains the actual class distribution as observed in the data, while the balanced training dataset randomly undersamples the majority classes (all classes have the same number of training instances).

In order to compare the impact of dataset size, we examined different stratified sampling approaches: i) *SAMPLE25* (random 25% stratified sample of the full dataset), ii) *SAMPLE10* (random 10% stratified sample of the full dataset), and iii) *SAMPLE1* (random 1% stratified sample of the full dataset). In all the experiments we applied 10-fold cross validation, using 90% of the instances for training and the remaining 10% for testing. Note that in the balanced datasets, undersampling of the training data of the majority classes is part of the cross validation, i.e. the test data is always unbalanced.

### 2.5.5 Classification Models

Considering the scale of the data as well as the features, we apply the following classifiers: i) *Naive Bayes* (a classifier that assumes that the likelihood of the features follows a Gaussian distribution), ii) *Logistic Regression* (a classifier that models the label probability based on a set of independent variables), iii) *Decision Tree* (a classifier that successively divides the features space to maximise a metric), and iv) *Random Forest* (a classifier that utilises an ensemble of uncorrelated decision trees).

### 2.5.6 Baselines and Multifeature Approach

While some related works deal with the prediction of unlinkable mentions [SWH14], no state-of-the-art baselines do exist which address the classification task proposed in our work. We follow the assumption that the ambiguity of a mention is strongly dependent on the available candidates in a KB as well as the mention length. These two features are known to strongly influence EL difficulty and have been used for creating gold standards of difficult test cases [HSN<sup>+</sup>12]. Thus, we consider the following baselines: i) CANDIDNUM (classification using only the feature  $m_{cand}$ ), and ii) MENTLENGTH

---

<sup>2</sup><http://l3s.de/~joao/SAC2019/>

(classification using only the feature  $m_{len}$ ). We compare the performance of these baselines with a MULTIFEATURE classifier which considers all the features described in Section 4.3 (cf. Table 2.1).

### 2.5.7 Configurations

Depending on the corpus (NYT in our case), some of the features need to be configured accordingly. For the *document topic* ( $d_{topic}$ ), we exploited the taxonomic classification provided by NYT. Each document was assigned to one of the following topics: *Arts, Automobiles, Books, Business, Education, Health, Home and Garden, Job Market, Magazine, Movies, New York and Region, Obituaries, Real Estate, Science, Sports, Style, Technology, Theatre, Travel, Week in Review, World, Miscellaneous*. For the *document publication age* ( $t_{age}$ ), we used *year* as the time interval. For the *mention's temporal document frequency* ( $t_{df}$ ), we used  $k=6$  months as the interval. For the *mention's semantics stability* ( $t_j$ ), we used  $K=50$  and  $\Delta=year$ , while in the Word2Vec Skipgram model we set the default setting as also used in [MSC<sup>+</sup>13] (300 dimensions, 5 words window size). Regarding the examined classifiers, we used their default configuration in WEKA [HFH<sup>+</sup>09].

### 2.5.8 Evaluation Metrics

To evaluate the performance of the different classifiers, we consider *Precision* (P) (the fraction of the correctly classified instances among the instances assigned to the class), *Recall* (R) (the fraction of the correctly classified instances among all instances of the class), and *F1 score* (the harmonic mean of P and R). We report the prediction performance per class as well as the macro average performance, to ensure that the size of each class has no impact on the representativeness of our metrics.

## 2.6 Evaluation Results

Below we show some results of the performance of different classification models followed by the analysis of the influence of the dataset size and we also show some results of the different features combinations.

### 2.6.1 Classification Performance

Table 2.2 summarises the overall results of the baselines (CANDIDNUM, MENT-LENGTH) and our multifeature approach (MULTIFEATURE) for the SAMPLE25 dataset. The table shows the macro averages of our performance metrics for both the unbalanced and balanced training dataset.

Table 2.2: Overall prediction performance (macro average) using SAMPLE25.

Method	Model	Unbalanced			Balanced		
		P	R	F1	P	R	F1
CANDIDNUM	NAIVE BAYES	0.38	0.35	0.34	0.37	0.40	0.32
	LOGISTIC REGR.	0.31	0.33	0.30	0.43	0.41	0.35
	DECISION TREE	0.74	0.47	0.50	0.48	0.61	0.47
	RANDOM FOREST	0.74	0.47	0.50	0.48	0.61	0.47
MENTLENGTH	NAIVE BAYES	0.25	0.33	0.29	0.36	0.42	0.26
	LOGISTIC REGR.	0.25	0.33	0.29	0.37	0.44	0.31
	DECISION TREE	0.25	0.33	0.29	0.42	0.47	0.40
	RANDOM FOREST	0.25	0.33	0.29	0.42	0.47	0.39
MULTIFEATURE	NAIVE BAYES	0.42	0.41	0.41	0.43	0.49	0.41
	LOGISTIC REGR.	0.45	0.36	0.35	0.43	0.50	0.40
	DECISION TREE	0.74	0.69	0.71	0.56	0.74	0.59
	RANDOM FOREST	<b>0.83</b>	<b>0.72</b>	<b>0.76</b>	<b>0.58</b>	<b>0.76</b>	<b>0.60</b>

In all cases, we observe that using the proposed MULTIFEATURE approach with a Random Forest classifier provides the best results, outperforming the baselines. Paired t-tests with  $\alpha$ -level 5% indicate that this improvement is statistically significant in all cases. With respect to the baselines, we observe that CANDIDNUM (number of mention’s candidate entities) outperforms MENTLENGTH (mention’s length). We also note that the unbalanced dataset achieves higher macro average F1 score compared to the balanced dataset (0.76 vs 0.60). In more detail, using the unbalanced training dataset we obtain higher macro average precision compared to the balanced dataset (0.83 vs 0.58), however recall is lower (0.72 vs 0.76).

Tables 2.4 shows the detailed performance per class for both the unbalanced and balanced training datasets. Looking at the MULTIFEATURE results of Random Forest for the unbalanced dataset, we notice that, as expected, the majority class EASY achieves high scores (0.92 precision and 0.97 recall). The MEDIUM class also performs very well (0.83 precision and 0.71 recall), while the HARD class achieves high precision (0.75) but lower recall (0.46). Regarding the HARD class, we see that using the balanced dataset recall is highly increased to 0.84, but precision drops to 0.21. We also observe that, when using MENTLENGTH with the unbalanced dataset, all classifiers learn to assign all instances to the majority class.

### 2.6.2 Influence of Dataset Size

Figure 2.3 shows the performance of our multifeature Random Forest classifier for different size of training data. As expected, the use of more training instances results in better performance. For instance, the F1 score using the unbalanced dataset increases from 0.65 (1% sample) to 0.7 (10% sample) and 0.76 (25% sample). We also noticed that the dataset size affects recall more than precision. In general, even when

Table 2.3: Prediction performance per class using SAMPLE25 with unbalanced training.

Method	Model	Unbalanced Training								
		Hard			Medium			Easy		
		P	R	F1	P	R	F1	P	R	F1
CANDIDNUM	NAIVE BAYES	0	0	0	0.37	0.11	0.17	0.78	0.96	0.86
	LOGISTIC REGR.	-	0	-	0.16	0.02	0.04	0.76	0.97	0.85
	DECISION TREE	0.67	0.08	0.14	0.73	0.35	0.47	0.83	0.98	0.90
	RANDOM FOREST	0.67	0.08	0.14	0.72	0.35	0.47	0.83	0.97	0.90
MENTLENGTH	NAIVE BAYES	-	0	-	-	0	-	0.76	1	0.87
	LOGISTIC REGR.	-	0	-	-	0	-	0.76	1	0.87
	DECISION TREE	-	0	-	-	0	-	0.76	1	0.87
	RANDOM FOREST	-	0	-	-	0	-	0.76	1	0.87
MULTIFEATURE	NAIVE BAYES	0.04	0.01	0.02	0.40	0.39	0.39	0.82	0.84	0.83
	LOGISTIC REGR.	0	0	0	0.58	0.11	0.18	0.78	<b>0.98</b>	0.87
	DECISION TREE	0.55	0.43	0.48	0.76	0.69	0.73	<b>0.92</b>	0.95	0.94
	RANDOM FOREST	<b>0.75</b>	<b>0.46</b>	<b>0.57</b>	<b>0.83</b>	<b>0.71</b>	<b>0.77</b>	<b>0.92</b>	0.97	<b>0.95</b>

Table 2.4: Prediction performance per class using SAMPLE25 with balanced training.

Method	Model	Balanced Training								
		Hard			Medium			Easy		
		P	R	F1	P	R	F1	P	R	F1
CANDIDNUM	NAIVE BAYES	0.06	0.32	0.10	0.26	0.03	0.05	0.80	0.86	0.83
	LOGISTIC REGR.	0.05	0.33	0.09	0.41	0.09	0.14	0.82	0.83	0.82
	DECISION TREE	0.10	0.65	0.17	0.43	0.48	0.45	0.92	0.69	0.79
	RANDOM FOREST	0.09	0.66	0.16	0.44	0.47	0.45	0.92	0.69	0.79
MENTLENGTH	NAIVE BAYES	0.05	0.72	0.08	0.14	0.12	0.13	0.88	0.43	0.58
	LOGISTIC REGR.	0.05	0.61	0.09	0.18	0.13	0.15	0.87	0.57	0.69
	DECISION TREE	0.06	0.40	0.10	0.33	0.38	0.35	0.87	0.64	0.74
	RANDOM FOREST	0.06	0.41	0.10	0.33	0.38	0.35	0.87	0.63	0.73
MULTIFEATURE	NAIVE BAYES	0.06	0.47	0.11	0.37	0.31	0.34	0.86	0.70	0.77
	LOGISTIC REGR.	0.07	0.49	0.12	0.33	0.39	0.36	0.88	0.63	0.74
	DECISION TREE	0.20	0.79	0.32	0.55	0.62	0.58	<b>0.95</b>	0.81	0.87
	RANDOM FOREST	<b>0.21</b>	<b>0.84</b>	<b>0.34</b>	<b>0.57</b>	<b>0.63</b>	<b>0.60</b>	<b>0.95</b>	<b>0.82</b>	<b>0.88</b>



using only 1% of the dataset, precision is quite high using the unbalanced dataset (0.78).

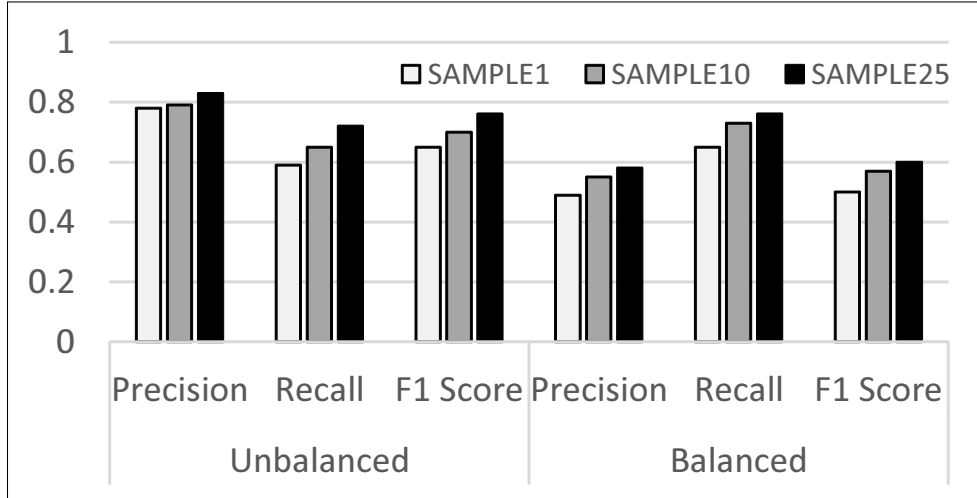


Figure 2.3: Influence of dataset size on prediction performance (macro average) using Random Forest.

### 2.6.3 Feature Analysis

To evaluate the usefulness of individual features, we compute the Mean Decrease Impurity (MDI) per feature, applied to the Random Forest model (the best performing classifier). MDI quantifies the importance of a feature by measuring how much each feature decreases the impurity in a tree, where in our analysis we considered information gain (entropy). We computed MDI using both the unbalanced and balanced SAMPLE25 datasets.

Figure 2.4 shows the average MDI score per feature (differences between the unbalanced and balanced datasets were minor). Surprisingly, the most useful feature is the mention’s normalised position ( $m_{pos}$ ), followed by the size of the sentence containing the mention ( $m_{sent}$ ), the frequency of the mention in the document ( $m_{freq}$ ), and the mention length ( $m_{len}$ ). We see that 3/4 of these features are related to the mention context. By inspecting several articles of the corpus we notice that a particular cause for this observation is the fact that author names are commonly added at the end of an article ( $m_{pos} \approx 1$ ). These entity mentions usually appear only once in the article ( $m_{freq} = 1$ ) and usually correspond to long-tail entities (with no Wikipedia entry). Hence such mentions tend to be of the HARD class. In addition, entities that appear early in the document (small  $m_{pos}$  value) are usually representative for the document, indicating more representative context which in turn facilitates their disambiguation. With regard to the high MDI score of  $m_{sent}$  (size of the sentence containing the mention), we noticed that several articles with HARD cases provide long lists of long-tail

entities (like the roster of a local team, or congress representatives). In such cases, the size of the sentence containing the mention is usually very small.

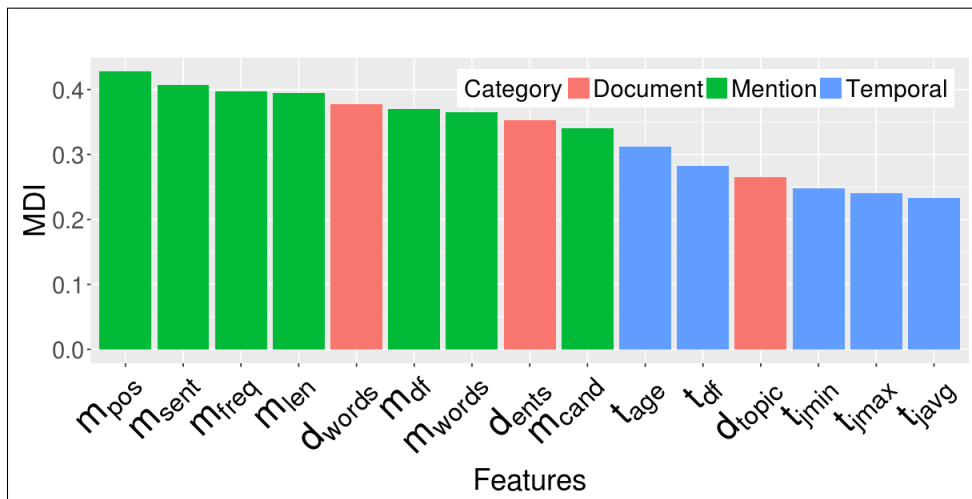


Figure 2.4: Attribute importance (Mean Decrease Impurity) per feature for SAMPLE25.

In general, we notice that the most important features are the mention-based features whereas temporal features impact the performance to a lesser extent (having though an MDI score of  $> 0.2$ ). With respect to the document-based features, the document size ( $d_{words}$ ) is the most useful (5th among all features), validating our hypothesis that small documents do not provide much context information and this hinders precise disambiguation of its mentions. With regard to temporal features, the publication age of the document containing the mention ( $t_{age}$ ) has the largest MDI value, while the three features related to the mention’s semantics stability ( $t_{jmin}$ ,  $t_{jmax}$ ,  $t_{javg}$ ) have the lowest contribution.

Note that a low MDI value indicates that, either the feature is not important or it is highly correlated with one or more of the other features. To assess correlation among the features, we examined the correlation matrix using Pearson’s correlation coefficient. The results are depicted in Figure 2.5 (we do not consider the nominal feature  $d_{topic}$ ).

The comparatively high correlation between the mention’s document frequency ( $m_{df}$ ) and temporal document frequency ( $t_{df}$ ) indicates that one of both likely is redundant, thus we can consider only  $t_{df}$  to avoid parsing the entire corpus. The high correlation among the min, max and average mention’s semantics stability ( $t_{jmin}$ ,  $t_{jmax}$ ,  $t_{javg}$ ) suggests that, in the case of our corpus, we may consider only one of these features. As expected, the number of mention’s characters ( $m_{len}$ ) is strongly correlated with the number of mention’s words ( $m_{words}$ ) (more words means longer strings), and the document size ( $d_{words}$ ) has a strong correlation with the number of document’s recognised entities ( $d_{ents}$ ) (large documents usually imply more recognised

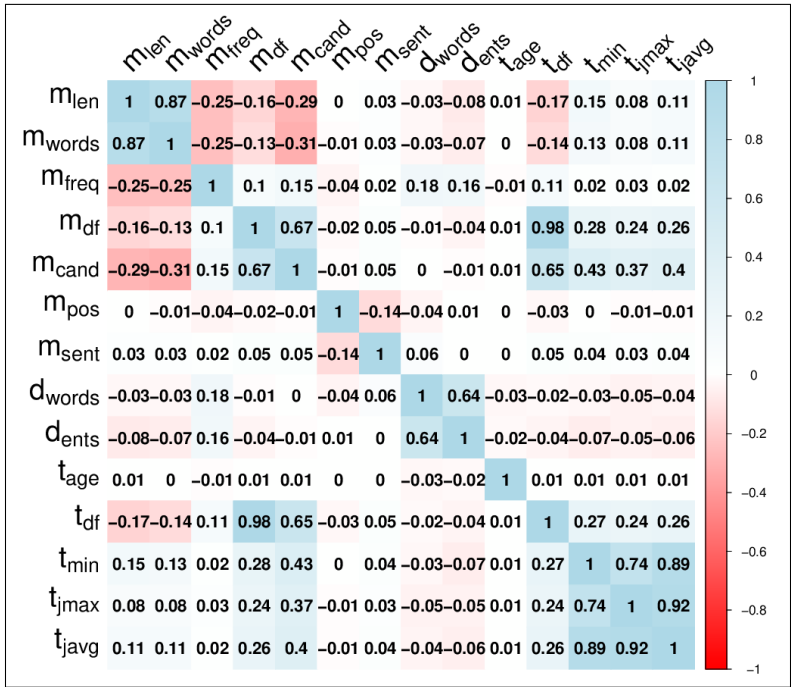


Figure 2.5: Correlation among features (Pearson’s r).

entities). An interesting correlation is that of the mention’s number of candidate entities ( $m_{cand}$ ) with the mention’s document frequency ( $m_{df}$ ) and temporal document frequency ( $t_{df}$ ). A possible explanation is the following: small values of  $m_{df}$  (or  $t_{df}$ ) may imply a less popular term which might correlate with a smaller amount of disambiguation candidates ( $m_{cand}$ ). This correlation may also explain the surprisingly low MDI value of  $m_{cand}$  (as shown in Figure 2.4).

We examined the performance of Random Forest without considering the features  $t_{df}, t_{jmax}, t_{javg}, m_{words}, d_{ents}$  (which are highly correlated to other features). Using SAMPLE25 and the unbalanced training dataset, we obtain the following macro average performance: P = 0.83, R = 0.71, F1 = 0.76. We observe that the results are almost the same with the ones reported for the entire feature set. Using the smaller SAMPLE1 dataset, we obtain P = 0.77, R = 0.58, F1 = 0.64. Again the performance is similar to the all-features approach (slightly worse). These results illustrate that we can omit some features that are expensive to compute and which have a strong correlation with other, less expensive features.

## 2.7 Impact on Entity Linking

To demonstrate the application of detecting difficult to link mentions, we assess the overall performance of semi-automated EL pipelines, where human annotators are guided by our classification task to complement system-generated entity links with

manual annotations in particularly challenging cases. We used three state-of-the-art EL systems (*Ambiverse*, *Babelify*, and *TagMe*), configured as described in the previous section (cf. Section 2.5.2) and using Wikipedia 2016 as the common reference KB. We consider a corpus for which gold standard annotations are provided, in particular the CoNLL-TestB ground truth [HYB<sup>+</sup>11], and applied the proposed method to generate difficulty labels.

From the commonly recognised mentions among the systems that also exist in the ground truth (2,471 mentions), we select a random set of  $N$  DIFFICULT mentions (labeled as *HARD* by our method) and consider that a human provides the correct link for these mentions. We do the same for a random set of  $N$  mentions predicted as *HARD* by a Random Forest classifier (PRED.DIFFICULT).<sup>3</sup> In both cases, if the number of *HARD* mentions is smaller than  $N$ , we fill up with random *MEDIUM* mentions. We compute the accuracy of the three systems (number of correctly linked mentions / total number of mentions) in both cases and compare the results with the accuracy of the systems on the same dataset before the human intervention (BEFORE), as well as with two baselines: i) one which randomly selects mentions for manual judgment (RANDOM), and ii) one which selects mentions based on their number of candidate entities, starting with the mentions having the more candidate entities (CANDIDATES). In all cases, for selecting the mentions to manually judge, we run the experiment 10 times for 10 different random sets of selected mentions, and we report the average results.

Figure 2.6 depicts the results for different proportion of manually judged entity links: 5% of the mentions ( $N = 124$ ) (left), 10% of the mentions ( $N = 247$ ) (middle), and 15% of the mentions ( $N = 371$ ) (right). We notice that the proposed method (DIFFICULT) highly improves the performance of all systems, while the improvement is considerably higher compared to the two baselines. *Ambiverse*, for instance, improves its accuracy from 0.81 to 0.84, 0.87, and 0.9, using 5%, 10%, and 15%, respectively, of the mentions for manual judgment. Moreover, using a pre-trained classifier (PRED.DIFFICULT), the improvement is again high and very close to the DIFFICULT case (outperforming again the two baselines). For example, *Ambiverse* improves its accuracy from 0.81 to 0.83, 0.86, and 0.88, using 5%, 10%, and 15%, respectively, of the mentions for manual judgment. These results demonstrate the effectiveness of our strategy on selecting difficult to link mentions (possible disambiguation errors).

---

<sup>3</sup>We trained the classifier using the full unbalanced training dataset of CoNLL and all features described in Section 4.3 apart from the three temporal features and the document topic (CoNLL does not provide this information).

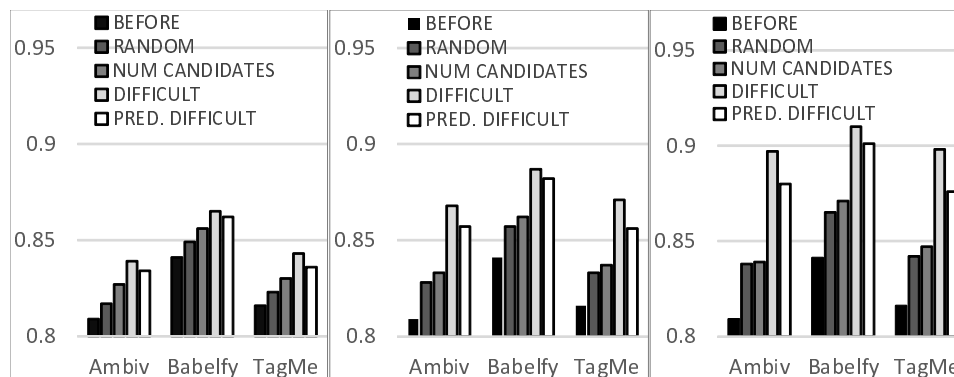


Figure 2.6: Effect of human feedback on the accuracy of semi-automated EL systems for different proportion of human judgments: 5% (left), 10% (middle), and 15% (right).

## 2.8 A Lightweight System for Entity Disambiguation

In order to support the disambiguation process for difficult cases as proposed in Chapter 2 a web-based system was created. The system was implemented as a webservice running in a local server that invokes the EL systems to parse the input text and calculate the mentions disambiguation disagreement among the employed EL systems. In Figure 2.7 it is possible to see the landing page where it is possible to type any free text in a textbox and once the button "Go" is clicked this webservice sends a request to the EL systems that perform the agreement calculation on-the-fly and displays the results right below in another textbox.

The textbox with the annotated text displays the original text plus the mentions that were recognized and disambiguated by the used EL systems. The mentions are highlighted with different colors to differentiate the difficulty level accordingly, i.e. mentions highlighted in green are classified as EASY, mentions highlighted in light blue are classified as MEDIUM and mentions highlighted in red are classified as HARD.

In parallel to this system that supports the disambiguation process for difficult cases, another tool was implemented to facilitate the disambiguation process. Figure 2.8 shows the user interface of this tool with one example of ambiguous mention identified in a snippet of text. The interface of this tool displays the input text on the top with the ambiguous mention highlighted in red. The basic idea behind this interface is to display the maximum information available to help the user decide upon the correct entity link for a given mention. It displays the candidate entities provided by different EL systems together with an image of the candidate entity and a snippet of text describing it. This small snippet of text is extracted from the entity's Wikipedia page. Moreover if the user wants to read more, it is possible to expand

WASHINGTON — As Lt. Col. Alexander S. Vindman sat in a stately chamber testifying on Tuesday, the White House posted on its official Twitter account a message denouncing his judgment. His fellow witness, Jennifer Williams, had barely left the room when the White House issued a statement challenging her credibility.

In President Trump's Washington, where attacks on his enemies real or perceived have become so routine that they now often pass unnoticed, that might not seem all that remarkable — but for the fact that Colonel Vindman and Ms. Williams both still work for the very same White House that was publicly assailing them.

Agreement Difficulty ▾

Go!

WASHINGTON — As Lt. Col. Alexander S. Vindman sat in a stately chamber testifying on Tuesday, the White House posted on its official Twitter account a message denouncing his judgment. His fellow witness, Jennifer Williams, had barely left the room when the White House issued a statement challenging her credibility. In President Trump's Washington, where attacks on his enemies real or perceived have become so routine that they now often pass unnoticed, that might not seem all that remarkable — but for the fact that Colonel Vindman and Ms. Williams both still work for the very same White House that was publicly assailing them.

Figure 2.7: Landing page for text input and agreement calculation.


the paragraph by hitting the *"more"* label or by clicking on the *"Wiki page"* label to be redirected to Wikipedia. Once the user has decided on the most appropriate candidate for the ambiguous mention, then it is possible to click the *"Select"* button.

In case none of the candidate entities is appropriate for the given mention, the user can either input manually the correct entity link on a text box that is shown on the right side of the page and click the *"Select"* button, or in case the user cannot find a reasonable entity link it is also possible to skip the current entity by clicking the *"Skip"* button. The expected output from this tool is a list of ambiguous mentions, the position where the mention appears in the input text, the candidate entities provided by each of the employed EL systems and the correct entity link selected by the user. The source code for these web-based and supportive tools is freely available<sup>4</sup>.

## 2.9 Conclusions and Future works


We have investigated the novel problem of detecting and understanding EL difficulty. To this end, we first introduced a method to generate difficulty labels for entity mentions in arbitrary corpora, by utilising agreement and disagreement sets obtained through state-of-the-art EL systems. As shown in the evaluation results, our approach to detect difficult to link mentions as part of a semi-automated EL pipeline can considerably improve the performance of state-of-the-art EL tools, by enabling the efficient prediction of critical cases which require manual labeling. For example, the accuracy of a popular EL system (Ambiverse) is increased by 6 percentage points

<sup>4</sup><https://github.com/renatosjoao/annotations>

 Please select the correct entity for the highlighted mention!

**Article Title:** Fittingly, Seles Has Final Say In 1991


**Article Content :** ...gs is worth boasting about, believes the fact that she can't do more than three push-ups per session is worth being embarrassed about, and finds no meaning in a final unless she wins it. No wonder that Seles, defending champion of the Virginia Slims Championships and a finalist in all 16 Kraft Tour events she entered this year, decided to render 1991 meaningful and comfortably forgettable by ending it as she began it at the Australian Open: with a major victory that earned her a brand-new black **JAGUAR**. And \$250,000. In a four-set final yesterday afternoon at Madison Square Garden, the 17-year-old Seles successfully defended her title and collected her 10th tournament victory. It came at the expense of Navratilova, the player she defeated in the United States Open final as she captured her third Grand Slam tournament of 1991. "Thank My Lucky Stars" "I just thank my lucky stars at how great of a year I've had," said Seles, "and then I want to forget about this year." In Navratilova's exp... [Read the full post](#)



**Jaguar Cars**

"Jaguar (<sup>i</sup>ˈdʒæɡjuː.ɑːr/ JAG-ew-ər) is the luxury vehicle brand of Jaguar Land Rover, a British multinational car manufacturer with its headquarters in Whitley, Coventry, England, owned by the... [+ more](#)


[Wiki page](#) [Select](#)



**Jaguar**

"The jaguar (*Panthera onca*) is a big cat, a feline in the Panthera genus, and is the only extant Panthera species native to the Americas. The jaguar is the third-largest feline after the tig... [+ more](#)

[Wiki page](#) [Select](#)



**Fender Jaguar**

"The Fender Jaguar is an electric guitar by Fender Musical Instruments characterized by an offset-waist body, a relatively unusual switching system with two separate circuits for lead and rh... [+ more](#)

[Wiki page](#) [Select](#)

**Other!**

Sometimes **None** of the displayed candidates is correct. If you know the correct entity for the mention in red, please add its Wikipedia page below.

#

[Select](#)

**Skip!**

If you are not sure or cannot find the correct entity!

[Skip](#)

Figure 2.8: Example of ambiguous entity mention (Jaguar).

when 10% of the recognised mentions, labeled as *HARD* by our method, are manually judged.

Subsequently, we introduced a set of features which can be used within a distantly supervised model for predicting difficult to link mentions on the fly, for cases where no labels can be assigned by the proposed labeling method or when real time analysis is needed. Evaluation results on the NYT corpus showed that difficulty labels can be predicted with high precision ( $>0.83$ ) and recall ( $>0.72$ ) even using limited amounts of training data, while recall can be further improved using a balanced training dataset. Our multifeature model highly outperforms baselines using the mention length or the number of mention's candidate entities only, demonstrating that context-specific features as well as temporal features are required in order to achieve reasonable performance. In addition, this prediction task can be used for detecting latent characteristics that affect EL performance on a given corpus. In the NYT corpus, for example, we saw that the position of the mention in the document characterises many *HARD* cases because long-tail entities (article authors) are usually listed at the last position.

Future work is concerned with reflecting more complex features, such as *lexical diversity* [DP02] or *document fluency* [HNF<sup>+</sup>16]. We also plan to investigate the effectiveness of common oversampling methods (like SMOTE [CBHK02]) as well as cost-sensitive classifiers and more balancing techniques, focusing on further increasing

the prediction performance for the minority class (*HARD*).



## Temporality of Prior Probability in Entity Linking

In this chapter I investigate the fact that an entity’s popularity is temporally sensitive and may change due to short term events. Thus EL tools should take into account the temporality factor when dealing with the disambiguation process for a given mention at different times.

### 3.1 Introduction

Entity linking is a well studied problem in natural language processing which involves the process of identifying ambiguous entity mentions (i.e persons, locations and organisations) in texts and linking them to their corresponding unique entries in a reference knowledge base. There has been numerous approaches and eventually systems proposing solutions to the task at hand. To mention a few, AIDA [HYB<sup>+</sup>11], Babelfy [MRN14], WAT [PF14] and AGDISTIS [UNR<sup>+</sup>14] for example, rely on graph based algorithms and the most recent approaches rely on techniques such as deep neural networks and semantic embeddings [HHJ15, ZSG16].

An important component in most approaches is the probability that a mention links to one entity in the knowledge base. The prior probability, as suggested by Fader et al. [FSEC09], is a strong indicator to select the correct entity for a given mention, and consequently adopted as a baseline. Computation of this prior is typically done over knowledge sources such as Wikipedia. Wikipedia in fact provides useful features and has grounded several works on entity linking [BP06, Cuc07, MC07, MW08, HYB<sup>+</sup>11].

An entity’s popularity is temporally sensitive and may change due to short term events. Fang and Chang [FC14] noticed the probability of entities mentioned in texts often change across time and location in micro blogs, and in their work they modeled spatio-temporal signals for solving ambiguity of entities. I on the other hand, take a macroscopic account of time, where perceivably a larger fraction of mention to entity

bindings might not be observable in the short time duration but are only evident over a longer period of time, i.e., over a year. These changes might be then reflected in a reference knowledge base and disambiguation methods can produce different results for a given mention at different times.

When using a 2006 Wikipedia edition as a reference knowledge base for example, the mention *Amazon* shows different candidates as linking destinations, but the most popular one is the entity page referring to *Amazon River*, whilst when using a 2016 Wikipedia edition, the same term leads to the page about the e-commerce company *Amazon.com* as the most popular entity to link to.

In this chapter, I systematically study the effect of temporal priors on the disambiguation performance by considering priors computed over snapshots of Wikipedia at different points in time. We also consider benchmarks that contain documents created and annotated at different points in time to better understand the potential change in performance with respect to the temporal priors.

I firstly show that the priors change over time and the overall disambiguation performance using temporal priors show high variability. This fact by itself strongly indicates that temporal effects should be not only taken into account in (a) building entity linking approaches, but have major implications in (b) evaluation design, when baselines that are trained on temporally distant knowledge sources are compared.

## 3.2 Problem Definition

In this section I briefly define the entity linking task as well as describe the methodology used in this chapter. Consider a document  $d$  from a set of documents  $D = \{d_1, d_2, \dots, d_n\}$ , and a set of mentions  $M = \{m_1, m_2, \dots, m_n\}$  extracted from  $d$ . The goal of the entity linking is to find a unique identity represented by an entity  $e$  from a set of entities  $E = \{e_1, e_2, \dots, e_n\}$ , with relation to each mention  $m$ . The set of entities  $E$  is usually extracted from a reference knowledge base  $KB$ .

A typical entity linking system generally performs the following steps: 1) mention detection which extracts terms or phrases that may refer to real world entities, and 2) entity disambiguation which selects the corresponding knowledge-base KB entries for each ambiguous mention.

Since the time effect is taken into account in the disambiguation task, I now pose entity linking at a specific time  $t$  as follows. Given a document  $d^t \in D^t$  and a set of mentions  $M = \{m_1, m_2, \dots, m_n\}$  from document  $d^t$ , the goal of the entity linking at time  $t$  is to find the correct mapping entity  $e^t \in E^t$  with relation to the mention  $m$ . The difference now is that the set of entities  $E^t$  is extracted from the reference knowledge base  $KB$  at different time periods.

Table 3.1: Information about the Wikipedia editions used for mining mention and entities. #Pages refers only to the number of entities’ pages, excluding special pages.

Year	Date	#Pages
2006	30/11/2006	~ 1.4 M
2008	03/01/2008	~ 1.9 M
2010	15/03/2010	~ 2.8 M
2012	02/09/2012	~ 3.5 M
2014	06/11/2014	~ 4.1 M
2016	01/07/2016	~ 5.1 M

### 3.2.1 Candidate Entities Generation and Ranking

As suggested by Fader et al. [FSEC09], the entity’s prior probability is a strong indicator to select the correct entity for a given mention. In this work’s case the entity’s prior probability is directly obtained from the Wikipedia corpus. To calculate entities’ probability, I parsed all the articles from a Wikipedia corpus and collected all terms that were inside double square brackets in the Wikipedia articles.  $[[\textit{Andy Kirk (footballer)} \mid \textit{Kirk}]]$  for instance, represents a pair of mention and entity where *Kirk* is the mention term displayed in the Wikipedia article and *Andy Kirk (footballer)* is the title of the Wikipedia article corresponding to the real world entity. In this way I created a list of mentions and possible candidate entities according to each Wikipedia snapshot used in this chapter’s experiments.

The probability of a certain entity  $e^t$  given a mention  $m$  was only calculated if the entity had a corresponding article inside Wikipedia at time  $t$ . Thus, the probability  $P(e^t|m)$  that a mention  $m$  links to a certain entity  $e^t$  is given by the number of times the mention  $m$  links to the entity  $e^t$  over the number of times that  $m$  occurs in the whole corpus at time  $t$ .

I created dictionaries of mentions and their referring entities ranked by popularity of occurrence for every Wikipedia edition as seen on Table 3.1. As an example of mention and its ranked candidate entities, in the KB created from the 2016 Wikipedia edition, the mention *Obama* refers in 86.15% of the cases to the president *Barack Obama*, 6.47% to the city *Obama, Fukui* in Japan, 1.79% to the genus of planarian species *Obama (genus)*, and so on and so forth.

I filtered out mentions that occurred less than 100 times for simplicity matters in the whole corpus and for every mention I checked whether the referring candidate entities pointed to existing pages inside the Wikipedia corpus at a given time, and only after these steps I calculated the prior probability values of the entities.

The proposed framework supports multiple selection of mention-entity dictionaries created from different *KBs* based on Wikipedia snapshots from different years.

### 3.3 Experiments

In this section I describe a set of experiments I performed to evaluate the temporality factor of the entities.

#### 3.3.1 Datasets

In order to evaluate the experiments I employed some data sets that are widely used benchmark datasets for entity linking tasks. *ACE04* is a news corpus introduced by Ratinov et al. [RRDA11] and it is a subset from the original ACE co-reference data set [DMP<sup>+</sup>04]. *AIDA/CONLL* is proposed by Hoffart et al. [HYB<sup>+</sup>11] and it is based on the data set from the CONLL 2003 shared task [SDM03]. *AQUAINT50* was created in the work proposed by Milne & Witten [MW08], and is a subset from the original AQUAINT newswire corpus [Gra]. *IITB* is a dataset extracted from popular web pages about sports, entertainment, science and technology, and health<sup>12</sup>, and it was created in the work proposed by Kulkarni et al. [KSRC09]. *MSNBC* was introduced by Cucerzan [Cuc07] and contains news documents from 10 MSNBC news categories. Table 3.2 shows more details about these datasets including the number of documents, documents' publication time, number of annotations as well as the reference knowledge base time.

#### 3.3.2 Prior Probability Changes

In many entity linking systems, the entity mentions that should be linked are given as the input, hence the number of mentions generated by the systems equals the number of entity mentions that should be linked. For this reason most researchers use accuracy to evaluate their method's performance. Accuracy is a straightforward

Table 3.2: *#Docs* is the number of documents. *Docs Year* is the documents' publication time. *#Annotations* is the number of annotations (Number of *non-NIL* annotations). *Annot. Year* is the reference KB time period where the annotations were taken from.

Dataset	#Docs	Docs Year	#Annotations	Annot. Year
ACE04 [RRDA11]	57	2000	257	2010
AIDA/CONLL [HYB <sup>+</sup> 11]	231	1996	4.485	2010
AQUAINT50 [MW08]	50	1998-2000	727	2007
IITB [KSRC09]	107	2008	12.099	2008
MSNBC [Cuc07]	20	2007	747	2006

<sup>1</sup><http://news.google.com/>

<sup>2</sup><http://www.espnstar.com/>

measure calculated as the number of correctly linked mentions divided by the total number of mentions.

Since this work takes into account the time variation, I only calculated accuracy over the total number of annotations that persisted across time, i.e. the entities from the ground truth that were also present in every Wikipedia edition used in this chapter’s experiments. Table 3.3 shows the accuracy calculated on the ground truth datasets using the prior probability model from different time periods. One can observe an accuracy change from 77.19% to 82.63% on *ACE04* using models created from Wikipedia 2006 and 2010 editions respectively, from 64.80% to 69.07% for *AQUAINT50* using models from 2006 and 2012 editions, from 64.13% to 68.16% for *AIDA/CONLL* using models from 2008 and 2014, from 46.60% to 49.76% on *IITB* using models from 2014 and 2006, and for *MSNBC* a change from 63.82% to 65.86% using models from Wikipedia 2012 and 2008 editions respectively

Even though it is out of the scope of this work to spot a temporal trend on the entities changes when using knowledge bases from different time periods, one can clearly see there is some temporal variability which is easily observed by the influence on the accuracy calculated over the ground truth datasets. A simplistic popularity only based method that takes into account reference *KBs* from different time periods can produce an improvement of 5.4 percentage points in the best case for the *ACE04* dataset and 2.0 percentage points in the worst case for *MSNBC* dataset.

### 3.3.3 Comparing Ranked Entities

I detected distinct changes when it comes to entity linking using Wikipedia as a knowledge base. The first case occurs when the entity page title changes but still refers to the same entity in the real world. For example in the 2006 Wikipedia edition the mention *Hillary Clinton* showed higher probability of linking to the referring entity page titled *Hillary Rodham Clinton* and in the 2016 Wikipedia edition, the same mention was most likely to be linked to the entity page titled *Hillary Clinton*. In this case only the entity page title changed but they both refer to the same entity in the real world.

The second case happens when an entity’s popularity actually changes over time.

Table 3.3: Accuracy of the models on different datasets across different time periods.

Dataset	2006	2008	2010	2012	2014	2016
ACE2004	77.19	81.17	<b>82.63</b>	80.96	80.54	79.49
AIDA/CONLL_testb	61.86	64.13	66.47	67.78	<b>68.16</b>	68.14
AQUAINT50	64.80	68.18	68.92	<b>69.07</b>	67.30	66.86
IITB	<b>49.76</b>	49.43	49.50	47.78	46.60	47.60
MSNBC	65.30	<b>65.86</b>	65.67	63.82	64.56	65.67

For example in the 2006 Wikipedia edition, the mention *Kirk* was most likely to be linked to the entity page titled *James T. Kirk* whereas in the 2016 Wikipedia edition the same mention showed a higher probability of linking to the entity page titled *Andy Kirk (footballer)*.

Another observation is the case when an entity mention that was considered unambiguous in the past and became ambiguous in a newer Wikipedia edition due to the addition of new information to Wikipedia. For example in the 2006 Wikipedia edition the mention *Al Capone* showed a single candidate entity, the north american gangster and businessman *Al Capone*, while in the newer 2016 Wikipedia edition, the same mention showed more candidate entities, including the former one plus a movie, a song, and other figures with the same name.

### 3.3.4 Top Ranked Entity Changes

Initially I was only concerned with the top ranked candidate entity for each mention. Thus I made comparisons between the dictionaries of mentions from Wikipedia editions 2006 and 2016 and despite the fact of observing 33,531 mentions in the 2006 version and 161,264 mentions in the 2016 version, only 31,123 mentions appeared in both editions. Moreover, when I took into consideration both the ambiguous and unambiguous mentions, in 9.44% of the cases the mentions changed their top ranked candidate entities, whilst when removing the unambiguous mentions this number increased to 15.36%. This is mainly due to the fact that most of the unambiguous mentions keep the same entity bindings, even though I spotted cases of mentions that were unambiguous and became ambiguous in a more recent knowledge base.

### 3.3.5 Top 5 Entities Changes

In another experiment I wanted to calculate the entities rank correlation. One way to calculate rank correlation for lists that do not have all the element in common, is to ignore the non conjoint elements, but unfortunately this approach is not satisfactory since it throws away information. Hence, a more satisfactory approach, as proposed by Fagin et al. [FKS03], is to treat an element  $i$  which appears ranked in list  $L_1$  and does not appear in list  $L_2$ , at position  $k+1$  or beyond, considering  $L_2$ 's depth is  $k$ . This measure was used to assess the changes in the top 5 candidate entities rank positions.

I calculated the rank correlation for 18,727 mentions, since this is the number of mentions that are ambiguous and appears both in the 2006 and 2016 Wikipedia corpus. I normalized the results so the values would lie between  $[0,1]$ . Any value close to 0 means total agreement while any value close to 1 means total disagreement. Thus I observed an average value of 0.59 with a variance of 0.05 and a standard deviation of 0.21. I noticed that in 71.98% of the cases the rank correlation values are greater than 0.5. That tells us there is some significant number of changes in the

candidate entities' rank's positions. Table 3.4 shows the mention *Watson* and its top 5 candidate entities together with their respective prior probabilities extracted from two different Wikipedia editions, one from 2006 and one from 2016.

Table 3.4: A mention example and its top 5 ranked candidate entities captured from two Wikipedia editions.

Mention	Entity	$P(e^t)$	Year
Watson	Doctor Watson	0.146	2006
	James D. Watson	0.130	
	Watson, Australian Capital Territory	0.115	
	Division of Watson	0.076	
	Watson	0.061	
Watson	Watson (computer)	0.068	2016
	Ben Watson (footballer, born July 1985)	0.054	
	Je-Vaughn Watson	0.050	
	Jamie Watson (soccer)	0.047	
	Arthur Watson (footballer, born 1870)	0.043	

## 3.4 Conclusions

In this work I conducted experiments with different Wikipedia editions and also created an entity linking model that uses the entity's prior probability calculated over different Wikipedia snapshots. One limitation of previous works is the fact that the systems are trained on a fixed time Wikipedia edition. An entity's prior probability is temporal in nature, and I have observed in my experiments that mention to entity bindings change over time. I could clearly see some temporal variability which should be taken into account for entity linking system's evaluations. As future work I plan to extend this chapter's experimental setup and build a ground truth for temporal entity linking as well as try to create an adaptive entity linking system.





## An Ensemble Learner for Combining Entity Linking Systems

In this chapter I investigate the viability of an ensemble learning approach. I propose a supervised model which predicts the "best-performing" model on a per-mention-basis in order to improve the disambiguation process.

### 4.1 Introduction

Entity linking (EL), or named entity recognition and disambiguation (NERD), is the task of determining the identity of entity mentions in text, thus linking a mention to an entity in a reference Knowledge Base (KB) like Wikipedia [SWH14]. For example, in the sentence "Jordan played for the Wizards", a typical EL system would link the term "Jordan" to the Wikipedia page of the basketball player *Michael Jordan* and the term "Wizards" to the Wikipedia page of the USA basketball team *Washington Wizards*.

EL is a crucial task of relevance for a wide variety of applications, such as information retrieval [RKC16], document classification [NXC<sup>+</sup>16], or topic modelling [CJY<sup>+</sup>16]. Usually, high precision and recall are required if EL results are to have a positive impact on any such application.

However, EL remains a challenging task. EL systems differ along multiple dimensions and are evaluated over different datasets [SWH14], while their performance varies significantly across domains and corpora [RUNN17]. For instance, evaluations using the GERBIL benchmark [RUNN17] have shown that the performance of EL systems is highly affected by the characteristics of the datasets, such as the number of entities per document, the average document length, or the salient entity types [URN15]. Thus, general-purpose EL remains a challenging task, where no single system has yet emerged as de-facto-standard across corpora and EL scenarios.

EL performance also varies strongly on each individual mention in the same cor-

pus. As is shown in the evaluation (Table 4.4), the F1 score of three established EL systems (TagMe, Ambiverse, Babelfy) on the popular CONLL dataset [HYB+11] ranges between 63.5% - 74.3% with an upper bound performance of 90.6% when selecting the most correct outputs of all three systems. This underlines that selecting the EL system on a per mention-basis rather than for a particular corpus, can significantly increase the EL performance. However, the selection of the most suitable system for a given mention remains a challenge. Prior works have shown that mentions which are difficult to link often share common characteristics [JFD19, HSN+12], which include ambiguity, indicated by a large number of candidates, mentions of long-tail entities which are not well represented in reference KBs, or mentions recognised in short documents with very limited context information.

Drawing on these observations, I argue that effective features can be derived from the corpus, the mention or the surface form to be linked, in order to predict the best-performing EL system on a per-mention-basis using supervised models. In this work I introduce an ensemble learning approach towards exploiting the EL capabilities of a set of ready-made EL systems not only for improving recall, but also to improve precision by predicting the most correct EL system considering the particular characteristics of each particular mention. I focused on exploiting ready-made (end-to-end) EL systems that are used as *black-box* systems using their default (suggested) configuration and without any corpus-specific training or tuning, because such systems are widely used in different contexts by also non-expert users.

Furthermore, I apply this approach to three established EL datasets and demonstrate significant performance improvements compared to both the individual EL systems and six baseline strategies. Specifically, when considering the largest dataset (CONLL), the proposed ensemble-based method significantly outperforms the best performing individual EL system by 10% of F1 score, as well as the top performing baseline by 5%.

In a nutshell, the following contributions are made:

- The introduction of the problem and a novel approach towards *Meta Entity Linking*, in short *MetaEL*, where outputs of multiple end-to-end EL systems are combined using an ensemble learning method for providing an improved set of entity links for a given corpus.
- I propose a diverse set of features which give suitable signals for predicting the EL system that can provide the correct link for a given mention, and build supervised classifiers which are used as part of an automated *MetaEL* pipeline.
- Using existing ground truth datasets and a set of three established and ready-made EL systems, I firstly provide detailed annotation and agreement statistics which demonstrate the potential performance improvement that an effective *MetaEL* method can provide. Then, I report the resulting EL performance gain of the proposed supervised approach as well as evaluation results on the

prediction task per se and the importance of the devised features, discussing also the limitations of this presented approach.

The rest of the chapter is organised as follows: Section 4.2 formulates the *MetaEL* problem and provides an overview of this approach. Section 4.3 details how supervised classification can be used for the problem at hand. Section 4.4 describes the evaluation setup. Section 4.5 reports the evaluation results. Section 4.6 discusses related works and the difference from the proposed approach. Finally, Section 4.7 concludes the chapter and discusses interesting directions for future research.

## 4.2 Problem definition

The current approach exploits a set of  $n$  EL systems  $(l_1, \dots, l_n)$  which operate on the same reference KB  $K$  and are applied to the same corpus  $D$ . The output is  $n$  sets of entity annotations  $\mathcal{A} = (A_1, \dots, A_n)$ , corresponding to  $n$  sets of entity mentions  $\mathcal{M} = (M_1, \dots, M_n)$ , each one produced by a different EL system  $l_i$ . The size of each set of entity mentions  $M_i$  can be different, since each system might have recognised different entity mentions.

**Definition 4.1** (Meta Entity Linking). *Assuming that, for a given corpus  $D$ , we have  $n$  sets of entity mentions  $\mathcal{M} = (M_1, \dots, M_n)$  and  $n$  sets of corresponding entity annotations  $\mathcal{A} = (A_1, \dots, A_n)$ , produced by  $n$  different EL systems  $(l_1, \dots, l_n)$ , the task of Meta Entity Linking, for short *MetaEL*, aims at providing a unified set of entity annotations  $A_u$ , where for each entity mention  $m \in (M_1 \cup \dots \cup M_n)$ , the most correct annotation is selected from  $(A_1 \cup \dots \cup A_n)$ .*

The solution proposed in this chapter, called *MetaEL+*, is based on *supervised classification*. Two variations are proposed, one focusing on high recall (LOOSE) and a more selective one focusing on high precision (STRICT). In both approaches, if at least two of the considered EL systems have recognised and disambiguated the same entity mention  $m$ , a *multi-label classifier* is used to predict which system to take into account. If only one EL system has recognised an entity mention, the STRICT approach predicts if the provided entity link is correct using a system-specific *binary classifier*. On the contrary, the LOOSE approach includes all annotations recognised by only one of the systems, thus focusing on high recall.

## 4.3 MetaEL+

This section describes the *features*, *classifiers* and *labeling* methods used by the proposed *MetaEL+* approach.

### 4.3.1 Features

I propose a set of features that can be easily computed for arbitrary corpora, i.e., I am not interested in features that, for example, require special metadata information about the documents. Inspired by related works on EL which study different factors that affect the performance of EL systems [SWH14, RUNN17], as well as by the observed characteristics of mentions that fail to be disambiguated correctly, I consider features of the following categories: i) *surface form-based* (features related to the word or sequence of words representing an entity), ii) *mention-based* (features related to the mention recognised in a document, in a specific position), and iii) *document-based* (features related to the document containing the mention). Below I detail each one of them.

### 4.3.2 Surface Form-based Features

**Number of words ( $s_{words}$ ):** the number of surface form's words. An EL system may perform better/worse on unigram surface forms that are usually more ambiguous than surface forms with more than one word.

**Frequency ( $s_f$ ):** the number of surface form's occurrences within the document. More occurrences implies that the document topic is closely related to the surface form, indicating more representative context to facilitate its disambiguation.

**Document frequency ( $s_{df}$ ):** the number of documents in the corpus  $D$  containing at least one occurrence of the surface form. Higher value implies popularity of the surface form, suggesting that more context is available about it which can facilitate its disambiguation by EL systems.

**Number of candidate entities ( $s_{cand}$ ):** the number of candidate entities in the reference KB (obtained by exploiting Wikipedia hyperlinks with anchor texts pointing to entities). Bear in mind that individual EL systems may perform better/worse on ambiguous mentions having a high number of candidate entities.

**Surface form's correct disambiguations per EL system ( $s_{corr}$ ):** number of times the surface form has been disambiguated correctly by a specific EL system on the given training dataset. An EL system which has disambiguated correctly a particular occurrence of a surface form is more likely to disambiguate correctly a different occurrence of the same term.

**Surface form's ratio of correct disambiguations per EL system ( $s_{ratio}$ ):** ratio of times the surface form has been disambiguated correctly by a specific EL system on the given training dataset. It is computed as the number of correct disambiguations divided by the sum of correct and wrong disambiguations. Similarly to the previous feature, the intuition is that an EL system which performed well on a number of occurrences of a particular surface form will perform well on the same term in the future.

### 4.3.3 Mention-based Features

**Mention’s normalised position** ( $m_{pos}$ ): the mention’s normalised position in the document, computed as the number of characters from the start of the document divided by the total number of characters in the document. Entities appearing early in the document are usually salient and representative for the document, indicating more representative context to facilitate their disambiguation.

**Mention’s sentence size** ( $m_{sent}$ ): the number of characters of the sentence containing the mention, specifically the length of the text between two punctuation marks containing the mention (considering only the punctuation marks “.”, “!”, “?”, “;”). Whereas an EL system may exploit the sentence containing the mention for disambiguating the entity, larger sentences may indicate more representative context for a particular mention.

### 4.3.4 Document-based Features

**Document size** ( $d_{words}$ ): the number of words of the document containing the mention. The document length may provide signals for EL system performance with some approaches being able to deal better with short documents (containing more concise but less context), while others with longer documents.

**Document’s recognised entities** ( $d_{ents}$ ): the total number of entities recognised in the document containing the mention. Given that EL systems tend to jointly disambiguate entities, some EL systems may perform better in the presence of a larger amount of recognised entities.

### 4.3.5 Classifiers

Since more than one EL system can provide the correct entity link for a recognised entity mention, the problem is posed as a *multi-label classification* task [TK07] where multiple labels (systems) may be assigned to each instance (entity mention). I experimented with a large number of different methods using the MEKA framework [RRPH16] (an open source implementation of several methods for multi-label classification), trying also different base classifiers for each method, including Naive Bayes (NB), Logistic Regression (LR), J48, Random Forest (RF), and Sequential Minimal Optimisation (SMO). In the evaluation section (Sect. 4.5), I report results only for the top performing method: *Binary Relevance* using *RF* as the base classifier.

As regards the case where only one EL system has recognised an entity mention  $m$ , a STRICT approach (as described in Sect. 4.2) needs to predict if the provided entity link is correct. For this,  $n$  binary classification models are needed, one for each considered EL system ( $l_1, \dots, l_n$ ), where the class label is either *true* (the EL system provides the correct entity link for  $m$ ), or *false* (the EL system does not provide the correct entity link for  $m$ ). I experimented with many different classification models,

including Naive Bayes (NB), Logistic Regression (LR), J48, Random Forest (RF), KNN and Sequential Minimal Optimisation (SMO). Here I report results only for SMO which consistently had the best performance across datasets.

### 4.3.6 Training and Labeling

For training supervised classifiers on the prediction tasks, one can generate training instances using manual labeling (e.g., from domain experts) [GHB<sup>+</sup>13], crowd-sourcing [BDR17], or existing ground truth datasets [RUNN17]. In my experiments, I make use of existing ground truth datasets (more in Sect. 4.4). After annotating the documents of the training corpus, I compute the feature values for each mention that exists in the ground truth and assign the corresponding class labels. For the *multi-label classifier*, I label the training instances by simply considering the systems that managed to correctly disambiguate the mention. For each *binary classifier*, I make use of only the annotations produced by the corresponding EL tool and label the training instances as either *true* or *false*.

## 4.4 Evaluation

I evaluated the EL performance of *MetaEL+* for a given set of ready-made EL tools. Since the previously introduced prediction task is an integral element of *MetaEL+*, I also evaluated the prediction performance of the proposed supervised classifiers.

### 4.4.1 Datasets

For training a supervised classifier, it is necessary to use datasets for which enough ground truth (GT) annotations are provided. I considered the following three datasets, each one containing at least 1,000 training annotations: i) *CONLL* (GT annotations for 1,393 Reuters articles [HYB<sup>+</sup>11]), ii) *IITB* (GT annotations for 107 text documents drawn from popular web pages about sports, entertainment, science, technology, and health [KSRC09]), iii) *NEEL* (GT annotations for > 9,000 tweets, provided by the 2016 NEEL challenge [CPPR<sup>+</sup>16]). Several other GT datasets have not been considered because of their very small size (e.g. *ACE2004*, *Aquaint*, *KORE50*, *Meij*, *MSNBC*). CONLL and NEEL are already split into training and test sets. For IITB the first 90% of the provided annotations was used for training and the remaining 10% for test (thus one can reproduce the results). In all datasets, GT annotations pointing to *NULL* or *OOKB* (out of knowledge base) are not taken into account. Table 4.1 shows the number of documents and annotations per dataset used for training and test (considering only the documents having at least one GT annotation).

Table 4.1: Ground truth datasets main statistics.

Dataset	#Train docs	#Train annots	#Test docs	#Test annots
CONLL	1,162	23,332	231	4,485
IITB	90	10,847	13	1,174
NEEL	3,342	6,374	291	736

#### 4.4.2 Entity Linking Tools

Three popular state-of-the-art EL tools were deployed: *Ambiverse* (previously AIDA) [HYB<sup>+</sup>11], *Babelfy* [MRN14], and *TagMe* [FS10]. These tools were selected because: i) they are end-to-end (ready-made) tools that can be easily used out-of-the-box, and ii) they are accessible through public APIs, thus one can directly use them. Moreover, they have been widely used in different contexts (each one having > 400 citations). Other EL systems, including more recent ones that make use of neural models, have not been considered because they do not satisfy these criteria. For *Ambiverse*, its default configuration was used. For *Babelfy*, the configuration suggested by the *Babelfy* developers<sup>1</sup> was used. For *TagMe* its default configuration and a confidence threshold of 0.2 to filter out low quality annotations was used.

Table 4.2: Performance of the used EL tools on CONLL [HYB<sup>+</sup>11]

System	Precision (%)	Recall (%)	F1 (%)
Ambiverse	80.7	64.7	71.8
Babelfy	81.5	68.2	74.3
TagMe	78.7	53.2	63.5

#### 4.4.3 Baseline and MetaEL+ Methods

Since the objective of *MetaEL* is the selection of output from multiple EL tools for achieving a better performance, each of the used tools (*Ambiverse*, *Babelfy* and *TagMe*) is considered a different and naive baseline. In addition, considering the agreement of the tools on the provided entity (majority vote) or their overall performance in a ground truth dataset, are two other predictive baselines [CANG16, SKH05]. As regards the *MetaEL* problem per se, [RP15] proposes a weighted voting scheme which ranks the candidate entities by considering the performance of the tools on a so-called ranking corpus (CONLL). The considered baselines are summarised below:

- Each considered EL system (**Ambiverse**, **Babelfy**, **TagMe**).
- **Random**: select one of the tools randomly.
- **Best System**: select the link provided by the system with the highest overall performance in the ground truth dataset.

<sup>1</sup>The configuration is available at: <https://goo.gl/NHXVVQ>

- **Majority+Random**: select the link provided by the majority of the tools. If all tools provide a different link, a random one is selected.
- **Majority+Best**: select the link provided by the majority of the tools. If all tools provide a different link, the system with the highest overall performance is selected. This method is similar to the *rule-based* method of [CANG16].
- **Weighted Voting**: the annotations are combined through the weighted voting scheme described in [RP15]. If the score is lower than the maximum precision for all annotators on the ranking corpus, the annotation is not considered.
- **Weighted Voting All**: the annotations are combined through the weighted voting scheme described in [RP15], however without filtering out annotations with a score lower than the maximum precision for all annotators.

The performance of the above mentioned baselines was compared with the following two *MetaEL+* approaches:

- **MetaEL+LOOSE**: a multi-label binary relevance classifier (with RF as the base classifier) is used when more than one tool provide a link for the same mention. I used the implementation and default configuration of MEKA 1.9.3 [RRPH16]. When more than one system is predicted, the *prediction confidence* scores provided by the classifier for each class are considered. In case of equal scores, the system with the highest overall performance in the training dataset is selected. If only one EL system has recognised a mention, I trust it and assign the entity provided by this system.
- **MetaEL+STRICT**: the same multi-label classifier from the MetaEL+LOOSE approach is used for cases where more than one tool provide a link for the same mention. However, this method is more selective: when a mention is recognised by only one EL system, a system-specific SMO binary classifier is used for predicting if the provided entity link is correct.

#### 4.4.4 Evaluating EL performance

The following metrics are used to evaluate the EL performance: **Precision (P)** (number of correctly disambiguated mentions divided by the number of recognised mentions), **Recall (R)** (number of correctly disambiguated mentions divided by the total number of not null annotations in the ground truth), and **F1 score (F1)** (harmonic mean of precision and recall).

#### 4.4.5 Evaluating the classification performance

Evaluation metrics for multi-label classification are inherently different from those used in single-label classification (like binary or multi-class) [TK07]. The results for the following metrics are reported: **Jaccard Index** (number of correctly predicted labels divided by the union of predicted and true labels), **Hamming Loss** (fraction of the wrong labels to the total number of labels), **Exact Match** (percentage of



samples that have all their labels classified correctly), **Per-class Precision** (P), **Recall** (R) and **F1 score** (if  $TL$  denotes the true set of labels for a given class and  $PL$  the predicted set of labels for the same class, then  $P = \frac{TL \cap PL}{PL}$ ,  $R = \frac{TL \cap PL}{TL}$ , and  $F1 = \frac{2 \cdot P \cdot R}{P + R}$ ).

Since the prediction of any of the tools that provides the correct entity is adequate, I also report the accuracy of the classifiers in each dataset when considering if the correct entity is provided by the predicted system. Based on this, I define **Real Prediction Accuracy** as the number of predictions for which the predicted system provides the correct entity divided by the total number of predictions.

Finally, for measuring the performance of the three binary classifiers used by the STRICT approach, I considered the **per-class P**, **R** and **F1 score** as well as the **macro-averaged F1 score**.

## 4.5 Results

Below I described the results obtained on a set of experiments that I performed to evaluate the proposed approach.

### 4.5.1 Annotation and agreement statistics

Table 4.3 provides detailed statistics about the annotations of the test datasets using the three EL tools. These statistics can help us better understand the characteristics of the datasets and the behavior of the considered tools.

Table 4.3: Annotation statistics of the test datasets.

	CONLL	IITB	NEEL
Total number of GT annotations:	4,485	1,174	736
Ambiverse annotations:	4,169	390	66
Babelify annotations:	9,578	868	246
TagMe annotations:	4,626	355	801
GT mentions recognised by 0/3 tools:	295 (6.6%)	694 (59.1%)	456 (62.0%)
GT mentions recognised by 1/3 tools:	468 (10.4%)	227 (19.3%)	225 (30.6%)
<i>Correct entity is provided:</i>	337 (72%)	63 (27.8%)	141 (62.7%)
GT mentions recognised by 2/3 tools:	1,251 (27.9%)	125 (10.6%)	43 (5.8%)
The 2 tools provide the same entity:	950 (75.9%)	88 (70.4%)	32 (74.4%)
The 2 tools provide different entities:	301 (24.1%)	37 (29.6%)	11 (25.6%)
<i>Correct entity is provided:</i>	1,061 (84.8%)	103 (82.4%)	37 (86%)
GT mentions recognised by 3/3 tools:	2,471 (55.1%)	128 (10.9%)	12 (1.6%)
3/3 tools provide the same entity:	1,786 (72.3%)	82 (64.1%)	8 (66.7%)
2/3 tools provide the same entity:	618 (25%)	37 (28.9%)	3 (25%)
Each tool provides a different entity:	67 (2.7%)	9 (7%)	1 (8.3%)
<i>Correct entity is provided:</i>	2,314 (93.6%)	119 (93%)	12 (100%)

The first four rows show the total number of GT annotations in each dataset and the number of annotations produced by each of the considered EL tools. The

next rows show the number of GT mentions recognised by zero, only one, two, or all three tools, as well as the number of mentions for which at least one of the tools provides the correct entity and the agreement of the tools in the provided entities. Different patterns across the datasets are noticed. Concerning CONLL, for instance, the majority of GT mentions were recognised by all three tools (55.1%), followed by mentions recognised by 2/3 tools (27.9%). On the contrary, for IITB and NEEL, the majority of GT mentions were not recognised by at least one system (59% and 62%, respectively). Based on these numbers, a high improvement of recall is expected when the annotations of the three tools are combined since a much larger number of GT annotations are expected to have been recognised by at least one system. Moreover, it is noticeable that for a quite high percentage of mentions recognised by only one EL tool, the provided entity is not correct (28% in CONLL, 72% in IITB, 37% in NEEL). Thus, an effective *MetaEL* method focusing on high precision should avoid including these annotations in the unified set of entity annotations.

With respect to the agreement of the tools on the provided entities, I noticed that when more than one system provides an entity for the same mention, the tools usually agree on the entity. Nevertheless for CONLL there is a high percentage of GT mentions where the tools disagreed and provided different entities (22% of all GT annotations). This percentage is 7% for IITB and around 2% for NEEL. The problem with these two datasets (especially with NEEL) is that the percentage of GT mentions recognised by zero or only one system is very high (78% for IITB and 93% for NEEL), in contrast to CONLL where the percentage is only 17%.

When only the GT mentions for which at least two tools provide a link are considered, then the percentage of mentions that need prediction is again high (26.5% for CONLL, 32% for IITB, and 27.3% for NEEL). For all these mentions, an effective *MetaEL* method needs to predict the system that can provide the correct entity link.

The above analysis shows that there is (i) a high percentage of mentions for which the EL tools provide different entities, and (ii) a high percentage of mentions for which no EL tool provides the correct entity. This means that predicting the system to consider or the correctness of an annotation can significantly improve the overall EL performance.

### 4.5.2 Upper bound performance

Given the GT of each dataset, one can compute the performance of an ideal *MetaEL* system that always makes a correct prediction (and thus no other method can provide better results). The first row in Table 4.4 shows the upper bound performance for each of the considered datasets. Comparing the upper bound performance for CONLL with the performance of the three individual tools on the same dataset (rows 2-4 in Table 4.4), *MetaEL* can highly increase the F1 score from 74.3% (of Babelfy, the top performing system) to 90.6%, i.e., >15 percentage points (or 22% increment). With respect to the other datasets, the F1 score of the upper bound performance

Table 4.4: Entity linking performance.

Method	CONLL-Test			IITB-Test			NEEL-Test		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
UPPER BOUND	100.0	82.8	90.6	100.0	24.3	39.1	100.0	25.7	40.9
AMBIVERSE	80.7	64.7	71.8	<b>85.2</b>	17.7	29.3	<b>76.6</b>	4.9	9.2
BABELFY	81.5	68.2	74.3	42.7	13.7	20.8	64.4	3.9	7.4
TAGME	78.7	53.2	63.5	72.3	14.9	24.7	67.5	23.4	34.7
RANDOM	79.3	74.1	76.7	52.7	21.6	30.6	64.3	24.5	35.4
BEST SYSTEM	80.3	75.0	77.5	57.9	<b>23.7</b>	33.6	65.7	<b>25.0</b>	<b>36.2</b>
MAJORITY+RANDOM	80.8	75.5	78.0	54.7	22.4	31.8	65.4	24.9	36.0
MAJORITY+BEST	80.5	75.3	77.8	57.7	23.6	33.5	65.7	<b>25.0</b>	<b>36.2</b>
WEIGHTED VOTING	80.8	72.5	76.4	44.8	17.3	25.0	63.5	22.7	33.4
WEIGHTED VOTING ALL	80.3	75.0	77.5	48.0	19.4	27.6	65.7	<b>25.0</b>	<b>36.2</b>
METAEL+LOOSE	84.8	<b>79.2</b>	<b>81.9</b>	57.7	23.6	33.5	65.7	<b>25.0</b>	<b>36.2</b>
METAEL+STRICT	<b>86.6</b>	75.2	80.5	84.8	22.3	<b>35.3</b>	73.0	9.9	17.5

is relatively low. As it is seen below, the reason is the low recall achieved by all tools. Nevertheless, the F1 score of the upper bound performance is much higher than that of the top performing individual system in each case (33% increment for IITB and 18% for NEEL). These results provide a good motivation for an effective *MetaEL* method that can achieve a high performance as close to the upper bound performance as possible.

### 4.5.3 Entity Linking Performance

Table 4.4 shows the EL performance of all approaches on the different datasets. The first row shows the upper bound performance and the next three rows the performance of the individual EL tools. The next six rows show the performance of the six baseline methods and the last two rows the performance of the two *MetaEL+* proposed methods.

To calculate the statistical significance of the presented results, I divided the test set of each dataset into 20 disjoint splits of equal number of annotations, and computed the F1 score on each split for each method (similar to the approach in [FC14]). Two-tail paired t-test was then applied to determine if the F1 scores of the methods and the baselines are significantly different.

Firstly, I noticed that the performance of the individual EL tools varies across datasets. As regards CONLL, Babelfy is the top performing tool and TagMe the tool with the worst performance (in terms of F1 score). For IITB, Ambiverse is the top performing tool and Babelfy the worst one. For NEEL, TagMe is the tool with the best performance and Babelfy the one with the worst performance. These results validate the initial motivation that the performance of EL systems varies across datasets.

Regarding the performance of the proposed MetaEL+ approaches, I noticed that the LOOSE approach achieves the highest F1 score in CONLL (the largest and most reliable dataset), outperforming the top performing individual system by 10% (from 74.3% to 81.9%) and the top performing baseline by 5% (from 78% to 81.9%). In more detail, recall of the top performing EL system (BABELFY) is improved from 68.2% to 79.2% (very close to the upper bound performance) and at the same time precision is improved from 81.5% to 84.8%. This is very promising given that, usually, improvement in recall affects precision negatively. With respect to the baseline methods, recall of the top performing baseline (MAJORITY+RANDOM) is improved from 75.5% to 79.2% and precision from 80.8% to 84.8%. All these improvements are statistically significant for  $\alpha$ -level = 0.05. It is also seen that, with a drop of recall to 75.2%, precision can be further improved to 86.6% using the STRICT approach. Here one would expect a higher improvement of precision, which means that the binary classifiers are not probably very effective in distinguishing *true* from *false* instances (this hypothesis is validated below).

In IITB, the METAEL+STRICT approach achieves the highest F1 score, outperforming the top performing EL tool (AMBIVERSE) by 20.5% and the top performing baseline by 5%. The proposed method combines a high recall (compared to that of the individual systems) with a very high precision (84.8%). Precision, in particular, is improved compared to the best baseline (BEST SYSTEM) by 46.5% while recall slightly drops from 23.7% to 22.3%.

Finally, in NEEL the LOOSE approach and four of the baseline systems achieve the same performance. This is not surprising given the very small number of cases that need prediction in this dataset (cf. Table 4.3). As regards the STRICT approach, it is noticeable that it highly improves precision from 65.7% (of the top performing baseline) to 73%, however with the cost of a high drop of recall (from 25% to almost 10%).

These results demonstrate that the proposed *MetaEL+* methods can significantly improve the performance of the individual systems and achieve results that are even competitive to recent EL systems that make use of neural models, like [CHLL18] and [KGH18] that report an F1 score of 80% and 82.4%, respectively, on the CONLL dataset.

#### 4.5.4 Prediction Performance

Table 4.5 shows the prediction performance of the multi-label classifier. It is seen that the *Jaccard Index* (ratio of correctly predicted labels) is high for CONLL (50.5%) and IITB (58.7%) but low for NEEL (36.5%). *Hamming Loss* (ratio of wrong labels) ranges from 26.9% (for IITB) to 41.3% (for CONLL). With respect to the most strict metric *Exact Match*, the score is 36.1% for CONLL, 54.0% for IITB, and 30% for NEEL. In general, it is noticeable that the classification performance is very good for IITB and satisfactory for CONLL. As it has already been stressed (cf. Sect. 4.4.5),

Table 4.5: Performance of multi-label classification

Evaluation metric	CONLL	IITB	NEEL
Jaccard Index (%)	50.5	58.7	36.5
Hamming Loss (%)	41.3	26.9	29.9
Exact Match (%)	36.1	54.0	30.0
Precision (%) of Ambiverse class	76.0	90.0	45.7
Precision (%) of Babelify class	81.8	86.4	23.1
Precision (%) of TagMe class	60.6	62.0	68.2
Recall (%) of Ambiverse class	60.4	46.0	44.4
Recall (%) of Babelify class	60.8	35.4	51.7
Recall (%) of TagMe class	61.9	35.4	26.2
F1 (%) of Ambiverse class	67.3	60.5	45.0
F1 (%) of Babelify class	69.8	50.2	31.9
F1 (%) of TagMe class	61.2	45.1	37.9
<b>Real Prediction Accuracy (%)</b>	91.1	95.9	69.6

these metrics evaluate the correct prediction of all class labels per instance. The *real prediction accuracy* (last row of Table 4.5) shows the classification performance when considering if the correct entity is provided by the predicted system. The score is more than 90% for CONLL and IITB, and almost 70% for NEEL. These results demonstrate the high performance of the proposed multi-label classifier.

Looking now at the per-class performance, for CONLL, the class label *Babelify* achieves the highest F1 score (69.8%) while the *TagMe* class has the lowest score (61.2%). On the contrary, in IITB the *Ambiverse* class achieves the highest F1 score (60.5%), due to its very high precision (90.0%), and *TagMe* the lowest (45.1%). In NEEL, the highest F1 score is again achieved by the *Ambiverse* class (45.0%), however the lowest by the *Babelify* class (31.9%). These results show that there is no class for which the classifiers have a consistent high performance.

#### 4.5.5 Binary classification

Table 4.6 shows the performance of the binary classifiers used by the METAEL+STRICT method. Firstly, it is important to highlight that the class distribution is very unbalanced. On average, around 78% of the annotations are correct (*true* class) and 22% are wrong (*false* class). This means that the *false* class is underrepresented, which makes the classification problem harder.

As expected, it is noticed that precision is very high for the majority *true* class in almost all cases, while recall is high for the minority *false* class. In CONLL, for

Table 4.6: Performance of binary classification.

<b>Evaluation metric</b>	<b>CONLL</b>	<b>IITB</b>	<b>NEEL</b>
Ambiverse - Precision (%) of <i>true</i> class	89.9	98.2	89.5
Ambiverse - Precision (%) of <i>false</i> class	25.5	21.4	31.0
Ambiverse - Recall (%) of <i>true</i> class	45.0	46.6	45.9
Ambiverse - Recall (%) of <i>false</i> class	79.0	94.4	81.8
Ambiverse - F1 (%) of <i>true</i> class	59.9	63.2	60.7
Ambiverse - F1 (%) of <i>false</i> class	38.6	34.9	45.0
Babelfy - Precision (%) of <i>true</i> class	91.5	96.5	93.8
Babelfy - Precision (%) of <i>false</i> class	27.2	66.3	51.7
Babelfy - Recall (%) of <i>true</i> class	52.2	30.2	51.7
Babelfy - Recall (%) of <i>false</i> class	40.4	79.5	66.7
Babelfy - F1 (%) of <i>true</i> class	66.5	46.0	66.7
Babelfy - F1 (%) of <i>false</i> class	38.6	34.9	45.0
TagMe - Precision (%) of <i>true</i> class	88.6	92.9	66.7
TagMe - Precision (%) of <i>false</i> class	30.5	35.5	32.3
TagMe - Recall (%) of <i>true</i> class	54.2	33.2	20.9
TagMe - Recall (%) of <i>false</i> class	74.3	93.5	78.3
TagMe - F1 (%) of <i>true</i> class	67.3	48.9	31.9
TagMe - F1 (%) of <i>false</i> class	43.3	51.4	45.8
Ambiverse - Macro-averaged F1 (%)	49.3	49.1	52.9
Babelfy - Macro-averaged F1 (%)	52.6	40.5	55.9
TagMe - Macro-averaged F1 (%)	55.3	50.2	38.9

example, precision of the *true* class ranges from 88.6% (TagMe classifier) to 91.5% (Babelfy), while that of the *false* class ranges from 25.5% (Ambiverse) to 30.5% (TagMe). On the contrary, recall of the *true* class ranges from 45% (Ambiverse classifier) to 54.2% (TagMe) and of the *false* class from 40.4% (Babelfy) to 79% (Ambiverse). Looking at the macro-averaged F1 scores, I noticed that their performance is close to 50% in almost all cases. TagMe classifier has the best performance in the two largest datasets (CONLL, IITB), however it has the worst performance in NEEL. It is evident from these results that there is much room for further improvement for binary classification.

#### 4.5.6 Feature Analysis

Table 4.7 shows the EL performance for different combinations of features when considering the largest ground truth dataset (CONLL) and the best performing *MetaEL+* method (METAEL+LOOSE).

With respect to the categories of features, the best performance is achieved when all categories are combined, which means that all contribute on achieving a high

performance. Regarding each individual category, the *surface form-based* features have the best performance, achieving an F1 score of 80.4%. The *mention-based* and *document-based* features achieve 77.6% and 78.2%, respectively. The best pair of feature categories is the *surface form-based* and *document-based* (81% F1) and the worst pair is the *mention-based* and *document-based* (77.6% F1). These results show that the *surface form-based* features have the highest contribution on achieving a good EL performance, and the *mention-based* features the lowest contribution.

Regarding the influence of each individual feature, the  $s_{ratio}$  (surface form’s ratio of correct disambiguations per EL system) has the highest effect when it is excluded, dropping the F1 score from 81.9% to 81%. The second most influential feature is  $m_{pos}$  dropping the F1 score to 81.1%, which means that the mention’s position in the document is a good indicator for the system that provides the correct entity.

Table 4.7: Effectiveness of different feature combination using METAEL+LOOSE on CONLL.

Features	P (%)	R (%)	F1 (%)
All features	<b>84.8</b>	<b>79.2</b>	<b>81.9</b>
Only surface form-based	83.2	77.7	80.4
Only mention-based	80.3	75.1	77.6
Only document-based	80.9	75.6	78.2
Surface form-based + mention-based	83.5	78.0	80.7
Surface form-based + document-based	83.8	78.3	81.0
Mention-based + document-based	80.3	75.1	77.6
All features except $s_{words}$	84.4	78.9	81.5
All features except $s_f$	84.0	78.5	81.2
All features except $s_{df}$	84.5	78.9	81.6
All features except $s_{cand}$	84.4	78.9	81.5
All features except $s_{corr}$	84.3	78.8	81.5
All features except $s_{ratio}$	83.8	78.3	81.0
All features except $m_{pos}$	83.9	78.4	81.1
All features except $m_{sent}$	84.1	78.6	81.3
All features except $d_{words}$	84.5	78.9	81.6
All features except $d_{ents}$	84.0	78.5	81.2

### 4.5.7 Synopsis and Limitations

The evaluation results can be summarised as follows:

- Combining multiple EL tools through a *MetaEL* approach can achieve a significantly better EL performance than individual systems in isolation.
- The proposed supervised ensemble approach (*MetaEL+*) significantly outperforms the individual EL tools and six baseline methods in the largest and most reliable datasets.

- A STRICT *MetaEL+* method which predicts if the entity provided by an EL system is correct can further improve precision without significantly affecting recall.
- The proposed multi-label classifier achieves a prediction accuracy of >90% in the two largest datasets of this work’s evaluation (CONLL and IITB), demonstrating its effectiveness.
- The proposed binary classifiers achieve a relatively low accuracy (F1 score  $\approx 50\%$ ), showing that there is much room for improvement of the STRICT *MetaEL+* method.
- All three categories of features contribute to achieving the highest performance. With respect to the individual features,  $s_{ratio}$  (surface form’s ratio of correct disambiguations by each EL system) and  $m_{pos}$  (mention’s normalised position in the document) seem to be the most influential features.

Limitations of the proposed work are mainly concerned with (i) the limited performance of the binary classifiers in the STRICT approach, and (ii) the need of corpus-specific training data.

## 4.6 Related Works

The survey in [SWH14] presents a thorough overview of the main approaches to EL, while more recent works (like [CHLL18], [KGH18] and [FCL<sup>+</sup>19]) exploit the idea of neural networks and deep learning. To the best of my knowledge, [RP15], [CANG16] and [CLT18] are the only previous works that focus on the related (yet different) problem of *MetaEL*, i.e., on how to combine the outputs of multiple EL tools for providing a unified set of entity annotations.

[RP15] proposes a weighted voting scheme inspired by the ROVER method [Fis97]. This method ranks the candidate entities by considering the performance of the systems on a so-called ranking corpus. Two of the baselines used in this work’s experiments consider this method. [CANG16] focuses on microposts and resolve conflicts by majority vote or, in the event of a tie, by giving different priorities to the annotations produced by each annotator. Two of the baselines in my work consider this approach. [CLT18] describes a framework to combine the responses of multiple EL tools which relies on the joint training of two deep neural models. However, this work is not applicable in the *MetaEL* problem since it makes use of external knowledge (pre-trained word embeddings and entity abstracts) as well as entity type information (a type taxonomy from each extractor), as opposed to the *MetaEL* task which only considers plain lists of entity annotations.

With respect to the related problem of *named-entity recognition* (NER), i.e., the detection of named entities in a given text and their classification in predefined categories like Person or Location, several works investigate how to combine the results of multiple NER methods [DKC<sup>+</sup>14, CANG16, SKH05, PRT16]. [DKC<sup>+</sup>14] tackles the



problem of concept extraction in *microposts* and proposes machine learning methods that make use of features describing the microposts for combining the results of different NER tools. [CANG16] also focuses on microposts and trains a multi-class SVM classifier. [SKH05] focuses on *bio-medicine* and proposes three methods for combining the results of various bio-medical NER systems: i) majority vote, ii) unstructured exponential model that considers the performance of the systems on training data, and iii) conditional random field that models the correlation between biomedical entities. I used the first two methods as baselines in the experiments. Finally, [PRT16] unifies the outputs of three different named-entity extraction models (dictionary, POS tagger, NER) in a specific order and merges the overlapping mentions.

A related line of research on the NER problem combines multiple classifiers through ensemble learning [WNC03, FIJZ03, SE13, SN14]. [WNC03] examined several stacking and voting (majority-based) methods that combine three different classifiers. In a similar way, [FIJZ03] combines the results of four classifiers, while [SE13] constructs an ensemble of seven classifiers. [SN14] evaluates the performance of 15 classification models, finding that ensemble learning can highly reduce the error rate of state-of-the-art NER systems. These works use as features the *predictions* of multiple supervised classifiers for deciding on the entity *type* of a given mention (from a pre-defined list of entity types), as opposed to the MetaEL task which combines *EL systems* and considers features extracted from the underlying *corpus* for training dedicated classifiers able to predict the *EL system* that can provide the correct link for a given mention.

A related interesting work is the NERD framework [RT11] which allows running multiple EL systems on the same text(s). NERD uses a common ontology for storing the results, thus providing a common representation format and facilitating the evaluation of NER and EL methods. However, it does not resolve conflicts like in the case of MetaEL. The work presented in this chapter can be used by this framework for conflict resolution and for providing a single set of entity annotations.

## 4.7 Conclusions

I have argued that the performance of entity linking (EL) on a given corpus may be optimised by combining the results of distinct EL tools. To this end, I introduced a novel approach towards *Meta Entity Linking (MetaEL)* where outputs of multiple end-to-end EL tools are unified on a per-mention basis through an ensemble learning approach. I modeled the problem as a supervised classification task and provided a rich set of features that can be used within a supervised classifier for predicting the EL system that can provide the correct entity link for a given mention.

Using existing ground truth datasets and three EL tools, I compared the performance of the proposed models with each individual EL tool and with six baseline methods. The results show that, considering the largest ground truth dataset

(CONLL), the multi-label classifier significantly outperformed the F1 score of both the best performing individual EL system (by 10%) and the best baseline (by 5%). Using binary classification for cases where a mention is recognised by only one EL system, a more selective (STRICT) approach that predicts the correctness of the provided entity link can further improve precision without significantly affecting recall. Results on the performance of the prediction tasks per se demonstrated the effectiveness of the proposed multi-label classifier. Finally, an extensive feature analysis showed that all the proposed features contribute on achieving a high EL performance.

Given the promising results of the experiments shown in this chapter, in the future it is planned to extensively evaluate the performance gain of MetaEL using different number and combinations of EL tools, including more recent tools that make use of neural models. This will provide a better understanding of the circumstances under which MetaEL has a significant effect in the EL performance. I also intend to study distantly supervised approaches where weakly labeled training data are automatically generated based on heuristics, aiming at solving the problem of obtaining corpus-specific training data. Finally, I plan to investigate the applicability of more advanced models for the binary classification task, in order to improve its (relatively low) performance.

## A Semantic Layer Querying Tool

### 5.1 Introduction

Web archiving has become an increasingly interesting field of research over the past decade [DBD<sup>+</sup>14, FHKN17, GC14, HNA17]. While, it was initially driven with the aim to preserve the web out of cultural needs, nowadays, an increasingly scientific interest is observed in diverse areas. As the web archives have become more widely known, several researchers began to investigate the potential and the limitations of such a resource as a complement to exploration of the currently active web [DMM<sup>+</sup>10]. Archiving the web is a complicated process that involves many tasks, including crawling and collecting the web pages to archive, defining efficient storage solutions and also providing an easy access to the documents. Exploring the archived documents still remains one of the greatest obstacles to providing the usability of web archives for non-expert users as well as a large variety of researchers.

In one exploratory search scenario where the user is unfamiliar with the documents content and how they are archived, a simple keyword-based search employed in traditional search engines does not work adequately and may produce results with poor quality. Thus, for exploring archived collections of documents, there is the need to go beyond the traditional keyword-based search and support more advanced strategies.

In a recent work by Fafalios et al. [FHKN17] the authors proposed to create a semantic layer that describes the information about the contents of documents in archived collections. With such a profile describing metadata information about each archived document one can benefit from the power of SPARQL language [Pru08] and run more advanced queries over a semantic layer and retrieve better quality results that answers complex queries such as: *"Find documents from 1995 discussing about lawyers in New York."*, *"Find documents mentioning soccer players born in Berlin."*, *"Find documents that mentions musicians who died before 1900."*, among others.

Despite the usefulness of such a semantic layer describing archived collections, one

main drawback of such approach is the need for manually writing structured SPARQL queries in order to search for information in a semantic layer. In this work I present a user-friendly and web-based search interface<sup>1</sup> that enables non-expert users to interact with the archived documents without the need to knowing how to formulate complex queries using the SPARQL language to answer basic information needs.

The main purpose of this work is to assist users in the expression of their information needs, in the formulation of their queries and understanding of their search results. The interface is mainly designed as a lightweight and simple search interface where the user can type free text keywords in an entry form and visualize the retrieved results in a list of results containing information about the retrieved documents. The source code of the search engine is made freely available online<sup>2</sup> and can be downloaded to be used with other semantic layers as long as the layers is created in RDF [BGM14] format according to the work proposed by Fafalios et al. [FHK17].

## 5.2 Related Works

In this section I describe some related works in the field of web archives as well as search engines designed specifically to operate on archived collections.

The Wayback Machine is a digital web archive that contains more than 525 billion web pages. In 1996 the Internet Archive<sup>3</sup> started to archive the internet itself and nowadays it is considered to be the biggest web archive in the whole world. Despite web pages, it also stores books, texts, audio and video recordings, as well as software programs. Anyone with access to a computer and internet can access older versions of a web page. The search mechanism allows users to retrieve content by URL or keywords and offers faceted navigation of the results page.

The Portuguese Web Archive (PWA) [GNMC09] preserves the Portuguese web, which is considered the web with most interest for the Portuguese community. It started officially in 2008 and it is also accessible from a public URL<sup>4</sup> allowing users to search either by full-text or by URL. From 1996 until 2007 the PWA has collected content mainly from the Internet Archive but soon after that, they began to make their own collections of the web.

The SolrWayback<sup>5</sup> is a web application for browsing historical gathered AR-C/WARC files from the Royal Danish Library<sup>6</sup>. Despite the traditional free text search it also offers interesting features such as the interactive link graph of domains, word cloud generation, the n-gram search visualisation and the possibility to search

---

<sup>1</sup><http://semanticlayers.l3s.uni-hannover.de/>

<sup>2</sup><https://github.com/renatosjoao/semanticlayersse.git>

<sup>3</sup><https://archive.org/>

<sup>4</sup><http://archive.pt>

<sup>5</sup><https://github.com/netarchivesuite/solrwayback>

<sup>6</sup><https://www.kb.dk/>

for images (i.e., by uploading an image the results are retrieved accordingly if that image has been collected in the past and from which domains).

Memento’s Time Travel service<sup>7</sup> enables users to search for versions of web pages that existed at some time in the past. These prior versions of web pages are named Mementos. The user provides the URL of the web page and a date of interest and Time Travel searches for Mementos in various web archives and version control systems. The results are returned in a list of Mementos, one per archive that actually holds one.

ArchiveWeb [FMN18] provides a keyword-based search system that returns results from archived collections as well as from the live web using the Bing search API.

Archive-It<sup>8</sup> is a web archiving service from the Internet Archive that enables organizations to build and preserve collections of web content. It offers search by URL, by metadata, and by keywords. Moreover, it also offers curation tools to control the extent, depth and description on the archived collections.

Jackson et al. [JLMR16] claim that the standard structure of a search engine results page (SERP), consisting of an ordered list of hits, is inadequate to support the needs of scholars and highlights the importance of the temporal dimension of web pages as well as issues surrounding metadata and veracity. Thus, in their work they implemented a search interface for web archives<sup>9</sup> which provides the results in two variants, the first one provides facets to filter the displayed results by several metadata values and the second one a “trends visualization” inspired by Google’s Ngram Viewer.

While existing systems offer interfaces to search through archived collections, they cannot satisfy more complex, but common information needs. I, thus design and develop a user-friendly search interface that can be used to search for information on semantic layers such as the one proposed in [FHKN17].

## 5.3 Open Web Archive Data Model

The model proposed by Fafalios et al. [FHKN17] is an RDF/S model that describes metadata and semantic information about the documents in a web archive. The full specification of the model is freely available online<sup>10</sup>. The root class `owa:ArchivedDocument` represents an archived document, and an archived document may be linked to some versions of the same document (i.e., instances of `owa:VersionedDocument`). Moreover, a versioned archived document can also be associated with some other important information like the date of first capture, the date of last capture and the total number of captures.

---

<sup>7</sup><http://timetravel.mementoweb.org/>

<sup>8</sup><https://archive-it.org/>

<sup>9</sup>[webarchives.ca](http://webarchives.ca)

<sup>10</sup><http://l3s.de/owa/>



Figure 5.1: Semantic Layers querying system architecture.

There are three main types of elements that can be associated with a versioned document. The first one includes metadata information like the date of the document publication, title of document and mime type format, the second one includes references to other web pages, and the third one is a set of entity annotations.

The RDF model of the semantic layer was created using an open source framework, called *ArchiveSpark2Triples*<sup>11</sup>, however it is not in the scope of this demonstration work to show how to create a semantic layer.

## 5.4 System Architecture

The user interface presented in this work is implemented as a java web application supported by a server-side component running the Apache Tomcat web server<sup>12</sup>. An overview of the system's architecture is shown in Figure 5.1.

The server component contains a running instance of the OpenLink Virtuoso<sup>13</sup> software with a semantic layer of the Occupy Movement 2011/2012 collection<sup>14</sup>. The Occupy Movement was known as a socio-political movement that expressed opposition to social and economic inequality around the world that began on September 2011.

OpenLink Virtuoso is a free and open source edition of Virtuoso Universal Server, a middleware and hybrid database engine that besides the traditional entity relationship database, it also supports object relational databases and RDF (Resource Description Framework) stores, enabling a database system to store and retrieve triples through semantic queries, for instance, by writing structured SPARQL queries.

<sup>11</sup><https://github.com/helgeh/ArchiveSpark2Triples>

<sup>12</sup><https://tomcat.apache.org/>

<sup>13</sup><https://virtuoso.openlinksw.com/>

<sup>14</sup><https://archive-it.org/collections/2950>

The advantage of the semantic layers approach is the fact that the document collection is not stored locally in the server, but only the semantic layer in the RDF format, and when the user issues a query, the running application models it automatically as a SPARQL query that depending on the user's information need it can also be forwarded to DBPedia<sup>15</sup> to assist responding the user's need. This forwarding step is done automatically via an important feature from SPARQL language known as federated query.

A federated query is basically the ability to take a query and provide answers based on information from many different sources (i.e., when the user issues a query in SPARQL language, it is capable of querying another SPARQL endpoint in real-time, without copying or moving data).

Finally the response is sent back to the user in a ranked list of documents surrogates containing the required information from the user's initial query.

## 5.5 Interface Design

This work represents the distillation of the authors' personal experiences working on web archives and the interaction with colleagues who always provided valuable feedback demonstrating a demand for more sophisticated exploration methods of archived collections in a lightweight and user-friendly interface.

The user interface of the Semantic Layers search landing page is presented in Figure 5.2. When the user starts typing any text in the search field, the entities available in the semantic layer are automatically suggested in a drop-down list displayed on top of the search field. In Figure 5.2 we can see that the entities *Barack Obama* and the next keyword being typed starts with *Bill* and the suggestion of entity *Bill Clinton* appears.


When the user enters terms in the search field and clicks the *Search* button, a query is sent to the server via the HTTP protocol, which interprets it and converts it into a SPARQL query. This query is then processed by the OpenLink Virtuoso server and the results are sent back to the user in the format of a search engine results page.

In the case of a query containing more than one entity, the user has the possibility to decide a priori if he wishes to obtain results showing documents that contains all the entities queried or at least one of them by selecting the options available below the search field. It is also possible to limit the search results to a pre-defined time window in case the user wants to filter documents by the crawling time.

For more complex queries there is the possibility to search documents containing some specific entity type (i.e., singer, journalist, lawyer, etc) or properties (e.g., birth place, college, nationality, etc) and location as well as a combination of entities and their attributes.

---

<sup>15</sup><https://wiki.dbpedia.org/>



## Semantic Layers

Barack Obama x

For example: Bill Clinton

Bill Clinton

A webpage must contain all the above entities.  
 A webpage must contain at least one of the above entities.

**Crawling date:**

Start Date

End Date

---

**Advanced Search**

**\_entity type**

For example: Basketball player, journalist, singer, etc.

**\_entity attribute**

For example: birth place, death place

**\_location**

For example: Europe, New York, Japan, etc

Search

Figure 5.2: Search page with query suggestion and advanced search.



The screenshot shows a search interface with a search bar containing 'Barack Obama' and a 'Search' button. Below the search bar, the results are displayed in a list format. At the top, it shows '42654 results (x.yz seconds)' and sorting options for 'Date', 'Language', and a list icon. The results list includes the following entries:

- The Prescott Daily Courier - Prescott, Arizona**  
1 Capture(s) [05/01/2012 -- 05/01/2012] 2012-01-05T16:34:09  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120105163409/http://www.dcourier.com/index.asp?TM=30552.22>
- Protestor | Alexander Higgins Blog**  
8 Capture(s) [20/12/2011 -- 10/08/2012] 2012-08-10T10:12:34  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120810101234/http://blog.alexanderhiggins.com/tags/protestor/>
- Occupy Binghamton | Facebook**  
4 Capture(s) [13/04/2012 -- 17/04/2012] 2012-04-17T05:05:22  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120417050522/https://www.facebook.com/OccupyBinghamton?filter=1>
- Vaccination | Alexander Higgins Blog**  
5 Capture(s) [21/12/2011 -- 10/05/2012] 2012-03-10T11:56:29  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120310115629/http://blog.alexanderhiggins.com/tags/vaccination/>
- American | Alexander Higgins Blog**  
9 Capture(s) [20/12/2011 -- 10/08/2012] 2012-08-10T09:29:48  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120810092948/http://blog.alexanderhiggins.com/tags/american/>
- Bradley Bailey | Facebook**  
1 Capture(s) [08/02/2012 -- 08/02/2012] 2012-02-08T03:42:54  
.. some snippet, more snippet, even more snippet  
<https://web.archive.org/web/20120208034254/http://www.facebook.com/people/Bradley-Bailey/100000038581568>

Figure 5.3: Results page with a list of surrogates.

The search results page of the proposed tool is presented in Figure 5.3. The results page is designed in a way that integrates navigation and search functionalities.

The results are displayed as a vertical list of surrogates that summarizes the retrieved documents. Each returned entry shows the title of the retrieved document, followed by the number of times the document was captured during the crawling process, the date referring to the first time a version of the document was captured, the date referring to the last time a version of the document was captured, the exact timestamp of the displayed document version, a snippet of the document's content as well as the electronic address (URL) where the document is archived in cases where the user wants to further analyse the document content.

In the search results page there is also the possibility to refine the results even more by selecting different time periods as well as to reorder the results that are being displayed according to other ranking criteria.

## 5.6 Querying the semantic layer

In this section it is demonstrated the Semantic Layers querying tool, a SPARQL query and how the user interface displays the retrieved results.

The SPARQL language offers advanced query capabilities, such as the federated query feature, which is the ability to issue one single query that is distributed to other SPARQL endpoints and provide the results in an aggregated solution.

By using the semantic layer querying tool to perform searches on the semantic layer one can infer knowledge related to the archived collection of documents that is very laborious to derive otherwise.

For example in the Listing 5.1 there is an example of how to issue a federated query in SPARQL language that searches for documents of the Occupy Movement collection containing the most cited journalists in the documents from the archived collections.

```

1 SELECT ?journalist (COUNT(DISTINCT ?page) AS ?num) WHERE {
2     SERVICE <http://dbpedia.org/sparql> {
3     ?journalist a yago:Journalist110224578 }
4     ?page a owa:ArchivedDocument ;
5     dc:hasVersion ?version .
6     ?version schema:mentions ?entity .
7     ?entity oae:hasMatchedURI ?journalist .
8 } GROUP BY ?journalist ORDER BY DESC(?num)

```

Listing 5.1: Federated SPARQL query to search for documents containing the most cited journalists.

The interface designed in this work is aimed at being simple and user-friendly, therefore, there is no need for the users to write complicated SPARQL queries. The user simply enters the desired keywords in the search fields accordingly and they are automatically translated into SPARQL. All the queries are then processed in the background and the user is presented only with the results page.

The results page that is presented to the user contains the documents ranked according to some previously defined criteria (i.e., ascending date of first capture). There is also the possibility to refine the results even more by selecting different time periods as well as to reorder the results that are being displayed according to different ranking criteria, for example, number of captures, publication date, first capture date, etc.

On the very top of the results pages the user can also re-submit queries with the possibility to add more entries to the original query.

## 5.7 Conclusions and Future Works

Searching for information in a web archive with billions of documents is a challenging task for a large variety of researchers as well as non-expert users, specially if the user is unfamiliar with the documents content and how they are archived.

The aim of designing a user-friendly search interface aligned with a semantic layer is to assist users to perform more advanced queries than simple keyword-based queries over archived collections. Moreover, despite the fact that a semantic layer already allows expert users to run sophisticated queries by exploiting the expressive power of the SPARQL language, such an interface can also assist non-expert users to search for documents that meet their information needs.

Future research includes the integration of several semantic layers into a single endpoint and also the deployment of different ranking models for the results returned by the SPARQL queries.



## Informative Tweet Identification

In this chapter I describe the viability of machine learning approaches for developing an automatic method to classify tweets according to their informativeness during catastrophic events.

### 6.1 Introduction

Lately Twitter has become an important channel for communication and information broadcasting. A large number of its users have been using the platform for seeking and sharing the information about events. Particularly, during undesired mass events like natural disasters or terrorist attacks, Twitter users post tweets, share updates, inform other users about current situations, etc. However, in addition to these information, a lot of tweets are merely for discussing and expressing opinions and emotions towards the events, which makes it challenging for professionals involved in crisis management to actually collect relevant information for better understanding the situations and respond more rapidly [VHSP10].

Considering the large volume of tweets published by Twitter users, manual sifting to find useful information is inherently impractical [Mei13]. Thus automatic mechanisms for identification of the informative tweets are required to assist not only the average citizen to become aware of the situation but also the professionals to take measures immediately and potentially save lives.

In this work, I investigated the viability of machine learning approaches for developing such an automatic mechanism. I studied both traditional ones that use handcrafted features, as well as the state of the art representation learning approach, the BERT-based models [DCLT19], to classify tweets according to their informativeness. Examples of *Informative* and *Not Informative* tweets from the CrisisMMD dataset [AOI18] are demonstrated in Table 6.1.

A rich set of features were designed and used with traditional machine learning

Table 6.1: Examples of tweets from CRISISMMD[AOI18] dataset.

<b>Tweet</b>	<b>Label</b>
<i>#SriLanka floods: 206 people dead, 92 still missing <a href="https://t.co/goLNqtiZUX">https://t.co/goLNqtiZUX</a> #top #news <a href="https://t.co/jJ9YCNSL4S">https://t.co/jJ9YCNSL4S</a></i>	<i>Informative</i>
<i>Thousands Homeless as Mexico Quake's Death Toll Tops 300 <a href="https://t.co/4iSf2hMv4m">https://t.co/4iSf2hMv4m</a> <a href="https://t.co/t28wYIcQoY">https://t.co/t28wYIcQoY</a></i>	<i>Informative</i>
<i>CR 218 bridge is closed, after Black Creek flooded during Hurricane Irma. Live at 5. @ActionNewsJax <a href="https://t.co/MDbNr7HnTh">https://t.co/MDbNr7HnTh</a></i>	<i>Informative</i>
<i>Glad to be alive. #lincoln #ford #Garmin #mkx #geico #HurricaneHarvey <a href="https://t.co/rgNfWHcnxo">https://t.co/rgNfWHcnxo</a></i>	<i>Not informative</i>
<i>@insideFPL it's been almost been 10 days.Please keep your promise. #frustrating #irma #fpl <a href="https://t.co/ISapDMh5Vl">https://t.co/ISapDMh5Vl</a></i>	<i>Not informative</i>
<i>i love huge murals!!!!!!! (by izak walter mora marambio) <a href="https://t.co/X4tPIG995y">https://t.co/X4tPIG995y</a></i>	<i>Not informative</i>

models and I also examined different neural embeddings. Furthermore, a hybrid model that leverages both the BERT-based models and the handcrafted features was proposed. All these models were evaluated on large datasets collected during several natural and man-caused disasters. In summary, the following contributions are made.

- The investigation of a rich set of features that include Bag-of-Words, text-based, and user-based features for traditional models. I studied the performance of BERT-based models for the informative tweet classification problem.
- I further proposed a hybrid model that combines a BERT-based model with handcrafted features for the problem.
- I conducted comprehensive experiments for evaluating the performance of these diverse models.
- Empirically, I demonstrated that deep BERT-based models outperform the traditional ones for the task without requiring complicated feature engineering, while the proposed model performs the best.

The remaining of this chapter is organized as follows. Firstly the related works are described in Section 6.2, then the methods and the features are presented in Section 6.3. Section 6.5 describes some experiments, datasets and give details about the proposed implementation methods. In Section 6.6 the results obtained from the experiments are reported. Finally, I draw some conclusions and point out some future directions in Section 6.7.

## 6.2 Related Works

Social media platforms such as Twitter and Facebook have become valuable communication channels over the years. Twitter enables people to share all kinds of information by posting short text messages, called tweets. Although social media services are full of conversational messages, it is also an environment where users post newsworthy information related to some natural or human-induced disaster. Identifying such information can help not only the ordinary citizen but it can also assist professionals and organizations in coordinating their response for potentially saving lives and diminishing catastrophic losses [ICDV15].

A number of automated systems have been proposed to extract and classify crisis related information from social media channels, for example CrisisTracker [RVT<sup>+</sup>13], Twitcident [AHH<sup>+</sup>12], AIDR [ICL<sup>+</sup>14], among others. For a more complete list of systems, please refer to the survey by Imran et al. [ICDV15].

Machine learning and natural language processing play an important role when it comes to classifying crisis related tweets automatically, and the approach applied to extract textual features can determine the performance of an automated classifier. Castillo et al. [CMP11] proposed automatic techniques to assess the credibility of tweets related to specific topics or events, using features extracted from user's posting behavior and tweet's text. Verma, et al. [VVC<sup>+</sup>11] used Naive Bayes and MaxEnt classifiers to find situational awareness tweets from several crises and Cameron et al. [CPRY12] described a platform for emergency situation awareness where they classified interesting tweets using an SVM classifier.

With the recent advances in natural language processing and the emergence of techniques such as word2vec [MCCD13, MSC<sup>+</sup>13] and GloVe [PSM14], deep neural networks have successfully been applied in similar tasks. Caragea et al. [CST16] for instance, demonstrated that convolutional neural networks outperformed traditional classifiers in tweet classification. Nguyen et al. [NAMJ<sup>+</sup>17] also used a convolutional neural network based model to classify crisis-relevant tweets. These results suggest a promising approach for this informative tweet classification task.

## 6.3 Methodology

Identifying informative tweets is a critical task, particularly during catastrophic events. There is however no simple rules that can be applied for the task. I therefore model the problem of informative tweets identification as a supervised learning problem. In the following subsections, I present a discussion about several models for the task. I start with some conventional classification models that make use of features engineered from the tweets as well as the users who posted the tweets. Next, I present the deep learning approaches for the task, and describe the proposed model.

### 6.3.1 Traditional models

Several machine learning approaches have been proposed for the task of automatically detecting crisis-related tweets, for example, Naive Bayes [LCCH18], Support Vector Machines [CST16], and Random Forests [KBR20]. Thus, as the baselines, I have trained these traditional classifiers to automatically classify a tweet into either *Informative* or *Not Informative*. Specifically, I have implemented the following models.

- **Logistic Regression (LR)** - a classifier that models the probability of a label based on a set of independent features,
- **Decision Tree (DT)** - a classifier that successively divides the features space to maximise a given metric (e.g., information gain),
- **Random Forest (RF)** - a classifier that utilises an ensemble of uncorrelated decision trees,
- **Naive Bayes (NB)** - a Gaussian Naive Bayes classifier,
- **Multilayer Perceptron (MP)** - a network of linear classifiers, (*perceptrons*) that uses the backpropagation technique to classify the instances, and
- **Support Vector Machine (SVM)** - a discriminative classifier formally defined by a separating hyperplane.

All the classifiers deployed in this work were implemented in Python using the machine learning library Scikit-Learn [PVG<sup>+</sup>11]. The source code of these models implementations is freely available at <https://github.com/renatosjoao/infotweets.git>.

### 6.3.2 Features

Inspired by previous works, I investigated a set of features based on the tweets' contents as well as on the users who posted the tweets [AR17, GRS<sup>+</sup>18, IEC<sup>+</sup>13, VVC<sup>+</sup>11]. These features are described as follows.

### 6.3.3 Text-based features

The ones that are calculated from the content of a tweet, including:

- $n_{chars}$ : This feature refers to the number of characters a tweet contains.
- $n_{words}$ : The number of words a tweet contains after removing symbols and patterns.



- $n_{hashtags}$ : The number of occurrences of #hashtags in a tweet. It can indicate the user wants to highlight some specific subject of interest.
- $n_{url}$ : The number of URLs contained in a tweet.
- $n_{at}$ : The number of @ tags in the tweet can be an indicator that the user is tagging people to draw their attention.
- $b_{hashtag}$ : Binary valued feature referring to the presence of #hashtags. True if at least one #hashtag is present in the tweet, false otherwise.
- $b_{at}$ : Binary valued feature referring to the presence of @ tags. True if the tweet contains @ tags, false otherwise.
- $b_{rt}$ : Binary valued feature referring to a retweeted message. True if the tweet contains retweet patterns, such as rt@, false otherwise.
- $b_{slang}$ : Binary valued feature referring to slangs in the tweet. True if the tweet contains any slang, false otherwise. Internet abbreviations are examples of text informality, which are representative of conversations. A dictionary of slangs was built from an online slang dictionary<sup>1</sup>.
- $b_{url}$ : Binary valued feature about the presence of URLs. True if at least one URL is present in the tweet, false otherwise.
- $t_{lex}$ : Tweet lexical diversity refers to the number of unique words divided by the total number of words in the tweet.
- $b_{interj}$ : Binary valued feature referring to an interjection. True if the tweet contains interjections, false otherwise. A dictionary of interjections was built from an online list of interjections<sup>2</sup>.
- $bow$ : Bag-of-Words features. Real-valued vectors are calculated with TF×IDF of the words and Twitter posts from each corpus for a finite number of words from the vocabulary.

## 6.4 User-based features

The ones that are calculated from the user who posted the tweet, including:

- $b_{usr}$ : Binary valued feature representing whether the user account is verified. True if the user has a verified account, false otherwise.

---

<sup>1</sup><https://www.lifewire.com/urban-internet-slang-dictionary-3486341>

<sup>2</sup><https://www.vidarholen.net/contents/interjections/>

- $n_{followers}$ : This feature represents the number of followers the user who posted the tweet has. Since this number may vary considerably this feature is calculated as  $\log_{10}(n_{followers} + 1)$ .
- $n_{followees}$ : Number of accounts the user who posted the tweet follows, calculated as  $\log_{10}(n_{followees} + 1)$ .
- $n_{tweets}$ : This feature represents the total number of tweets posted by the user. There can be the case where the user has not posted many tweets as well as there can be cases of influential users who post messages more frequently, thus this feature is calculated as  $\log_{10}(n_{tweets} + 1)$ .

### 6.4.1 Deep learning approaches

Now I discuss deep learning based approaches that are widely used in recent works [NAMJ<sup>+</sup>17, NCC18].

### 6.4.2 Word embedding methods

The traditional models such as the *Bag-of-Words* do not capture well the meaning of the words and consider each word as a separate feature. Word embeddings have been proposed and widely used neural models that map words into real number vectors such that similar words are closer to each other in a higher dimensional space. The word embeddings captures the semantical and syntactical information of words taking into consideration the surrounding context.

In this work, I examine the following typical word embedding methods:

- **Word2vec** [MSC<sup>+</sup>13] is one famous method of neural words embeddings initially proposed in two variants: (i) a Bag-of-Words model that predicts the current word based on the context words, and (ii) a skip-gram model that predicts surrounding words given the current word.
- **GloVe** is an extension to the Word2vec method for efficiently learning word vectors, proposed by [PSM14] which uses global corpus statistics for words representations and learns the embeddings by dimensionality reduction of the co-occurrence count matrix.
- **Fasttext** [BGJM16] is an extension to the skip-gram model from the original Word2vec model which takes into account subword information, i.e. it learns representations for character n-grams, and represents words as the sum of the n-gram vectors. The idea is to capture morphological characteristics of words.

I make use of the pre-trained word vectors of the above models<sup>3, 4, 5</sup>. The feature vector of each tweet is then determined by taking the average of all embedding vectors of its words.

### 6.4.3 Text embedding methods

Generalized from word embeddings, text embedding methods compute a vector for each group of words taken collectively as a single unit, e.g., a sentence, a paragraph, or the whole document. In this work, I examine a typical method for text embedding, namely Doc2vec, and state-of-the-art ones, namely BERT-based models.

- **Doc2vec** generates efficient and high quality distributed vectors of a complete document [MSC<sup>+</sup>13]. The main objective of Doc2Vec is to convert the sentence (or paragraph) into a vector. It is a generalization on the Word2vec model.
- **BERT** is a model developed on a multi-layer bidirectional Transformer encoder [VSP<sup>+</sup>17, DCLT19]. It makes use of an attention mechanism that learns contextual relations between words in texts. In its generic format, the Transformer includes two separate mechanisms, an encoder that reads input text and a decoder that produces the task prediction. The encoder is composed of a stack of multiple layers, and each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The decoder is also composed of a stack of multiple identical layers with the addition of a third sub-layer, which performs multi-head attention over the output of the encoder stack. One key component of the Transformer encoder is the multi head self-attention layer, i.e. a function that can be formulated as querying a dictionary with key-value pairs.

The most straightforward usage of BERT is to employ it as a blackbox for feature engineering. This is the combination of the default BERT model and conventional classifiers. The final hidden state of the first word ([CLS]) from BERT is the encoded sentence representation and it is input to conventional classifiers for the predictions task.

The original BERT model is pre-trained in a general domain corpus. Thus, for a text classification task in a specific domain, the data distribution may be different. In this way in order to obtain improved results, we need to further train BERT on a domain specific data. There are a couple of ways to further train BERT on a domain specific corpus. The first one is to train the entire pre-trained model on the new corpus and feed the output into a softmax function. In this way, the error is back

---

<sup>3</sup>Word2vec: <https://code.google.com/archive/p/word2vec/>

<sup>4</sup>GloVe: <https://nlp.stanford.edu/projects/glove/>

<sup>5</sup>Fasttext: <https://fasttext.cc/>

propagated throughout the entire model's architecture and the weights are updated for this domain specific corpus. Another method is to train some of BERT's layers while freezing others, or we can freeze all the layers and attach extra neural network layers and train this new model where only the weights of the attached layers will be updated. These are so called fine tuning procedures, and in this work I will be fine tuning BERT, by encoding Twitter sentences with the BERT encoder and running more training iterations and backpropagating the error throughout the entire model.

#### 6.4.4 BERT

BERT is a model developed on a multi-layer bidirectional Transformer encoder [VSP<sup>+</sup>17]. It makes use of an attention mechanism that learns contextual relations between words in texts. In its generic format, the Transformer includes two separate mechanisms, an encoder that reads input text and a decoder that produces the task prediction. BERT's architecture (encoder and decoder) is shown in Figure 6.1.

The encoder is composed of a stack of N layers, and each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The decoder is also composed of a stack of N identical layers with the addition of a third sub-layer, which performs multi-head attention over the output of the encoder stack. One key component of the Transformer encoder is the multi head self-attention layer, i.e. a function that can be formulated as querying a dictionary with key-value pairs.

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) \cdot V, \quad (6.1)$$

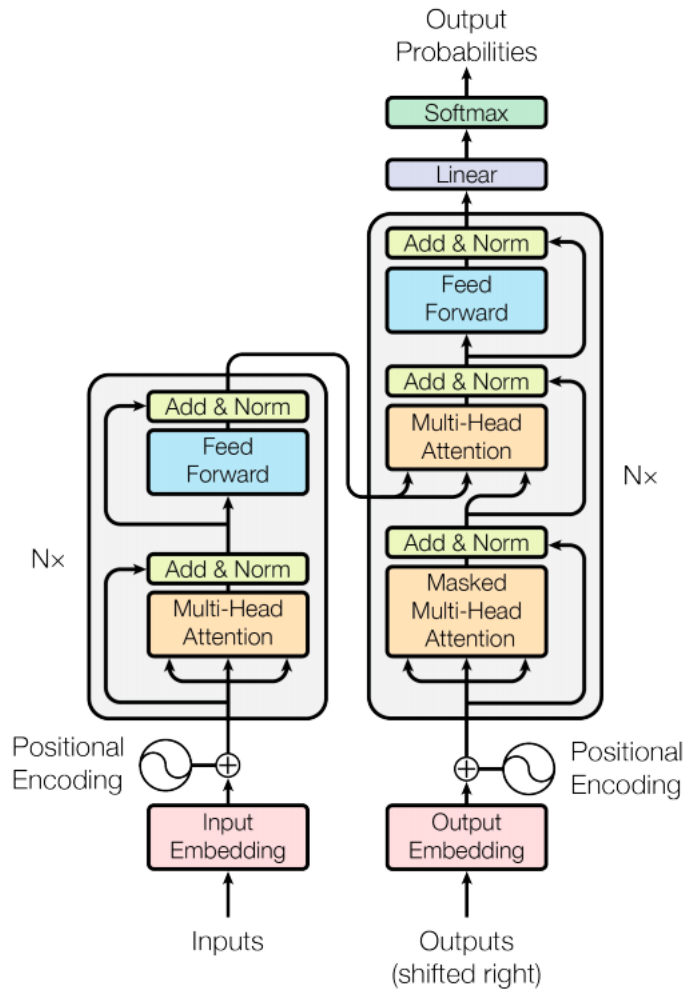
$$\text{where } Q \in \mathbb{R}^{n_q \cdot d_k}, K \in \mathbb{R}^{n_e \cdot d_k}, V \in \mathbb{R}^{n_e \cdot d_v} \quad (6.2)$$

where Q is a matrix of queries, K and V are matrixes of keys and values respectively.

The original BERT model is pre-trained in a general domain corpus. Thus, for a text classification task in a specific domain, the data distribution may be different. In this way in order to obtain improved results, one need to further train BERT on a domain specific data.

There are a couple of ways to further train BERT on a domain specific corpus. The first one is to train the entire pre-trained model on the new corpus and feed the output into a softmax function. In this way, the error is back propagated throughout the entire model's architecture and the weights are updated for this domain specific corpus. Another method is to train some of BERT's layers while freezing others, or we can freeze all the layers and attach extra neural network layers and train this new model where only the weights of the attached layers will be updated. These are so called fine tuning procedures, and in this work I will be fine tuning BERT, by encoding Twitter sentences with the BERT encoder and running more training iterations and backpropagating the error throughout the entire model.

Figure 6.1: BERT's architecture.



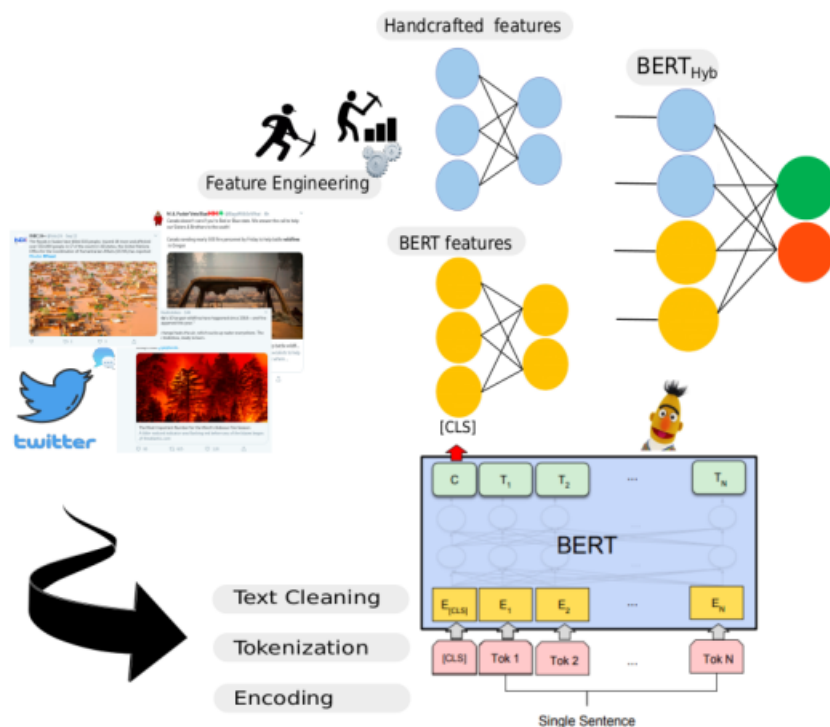
### 6.4.5 Dropout layers

Overfitting is an issue in machine learning of serious concern. It happens when a network classifies the training dataset effectively but fails to produce satisfactory performance results on unseen data, usually the test set. Generally this behavior happens when a neural network is built and the neurons start to detect the same features repeatedly. One way to address this situation is to employ dropout layers [SHK<sup>+</sup>14] where they randomly disconnect the connections between some neurons with a certain dropout rate. Dropout can be interpreted as a way of regularizing a neural network by adding noise to its hidden units. In this manner the network is able to generalize better and produce more efficient results on unseen data.

### 6.4.6 A Hybrid BERT model

I now describe a hybrid model, called  $BERT_{Hyb}$ , that combines both the handcrafted features with the ones learned by BERT. Figure 6.2 shows an overview of the model architecture:  $BERT_{Hyb}$  model feeds a vector of handcrafted features from the tweet through a linear layer, and also feeds the vector produced by BERT for the first token (CLS) of the tweet through another linear layer. The outputs of these two layers are concatenated and fed through a third linear layer, whose output is subsequently fed through a softmax layer to produce the prediction whether a tweet is *Informative* or *Not Informative*.

Figure 6.2: BERT’s hybrid model architecture.



## 6.5 Evaluation Setup

I now present the experiments to empirically evaluate the methods presented above. In the following subsections, I shall describe the datasets, define the evaluation metrics, the experiment settings, and report the results.

### 6.5.1 Datasets

I use the following datasets to evaluate the models.

Table 6.2: Complete datasets classes distributions.

<i>Dataset</i>	<i>#Informative</i>	<i>#Not Informative</i>	<i>Total</i>
COVID	3,772	4,221	7,993
CRISISLEX T6	32,461	27,620	60,081
CRISISLEX T26	16,849	7,731	24,580
CRISISMMD	11,509	4,549	16,058

Table 6.3: Subsets classes distribution.

<i>Dataset</i>	<i>#Informative</i>	<i>#Not Informative</i>	<i>Total</i>
COVID <sub>SUBSET</sub>	3,378	3,816	7,194
CRISISLEX T6 <sub>SUBSET</sub>	20,568	17,422	37,990
CRISISLEX T26 <sub>SUBSET</sub>	11,023	4,442	15,465
CRISISMMD <sub>SUBSET</sub>	9,343	3,443	12,786

- **CrisisLexT26** [OVC15] - This is a dataset of tweets collected during twenty six large crisis events in 2012 and 2013, with about 1,000 tweets labeled per crisis for informativeness, information type, and source.
- **CrisisLexT6** [OCDV14] - This dataset includes English tweets posted during six large events in 2012 and 2013, with about 60.000 tweets labeled by relatedness as *On-topic* or *Off-topic* with each event. I assume the tweets labeled as *On-topic* being the *Informative* tweets and *Off-topic* being *Not Informative* respectively.
- **CrisisMMD** [AOI18] - CrisisMMD is a dataset that contains tweets with both text and image contents. There are 16,000 tweets that were collected from seven events that took place in 2017 in five countries.
- **Covid** [NVR+20] - This dataset consists of 10K English Tweets collected during the Covid pandemic. It is split into training set with 3303 *Informative* tweets and 3697 *Uninformative* tweets, and a validation set with 472 and 528 *Informative* and *Uninformative* tweets respectively.

In their original form, the above datasets provide only tweets' content together with their ids and labels. To calculate the user based features I crawl from Twitter the full information of all the tweets. However, some tweets are no longer available. I thus create a version of each dataset that consists of the subset of tweets that I can crawl full information from Twitter. These versions are COVID and COVID<sub>SUBSET</sub>, CRISISLEX T6 and CRISISLEX T6<sub>SUBSET</sub>, CRISISLEX T26 and CRISISLEX T26<sub>SUBSET</sub>, CRISISMMD and CRISISMMD<sub>SUBSET</sub> respectively. The basic statistics of all the datasets and their subsets are shown in Tables 6.2 and 6.3 respectively.

### 6.5.2 Evaluation Metrics

To evaluate the informative tweets classification task I employ the following performance metrics. **Precision (P)**: the fraction of the correctly classified instances among the instances assigned to the class. **Recall (R)**: the fraction of the correctly classified instances among all instances of the class and **F-score (F1)**: the harmonic mean of precision and recall. In this work I compute the metrics independently for each class and then take the average, i.e. Macro Precision, Macro Recall and Macro F-score.

### 6.5.3 Experiment settings

I normalized all characters in the tweets to their lower-cased forms followed by the removal of punctuation and non ASCII characters as well as non English words, then I calculated the text-based features and user-based features. The Bag-of-Words feature was calculated for the entire corpus of tweets, however in the experiments I only calculated it for words appearing at least 5 times in the entire corpus and up to a limit of 10000 times. The words with length less than two characters were also pruned.

In parallel I then tokenized the sentences and encoded the tokens using the BERT encoder. Each dataset is randomly split into 10 mutually exclusive subsets and 10-fold cross validation was used to measure the performance of the models. For the conventional classifiers I used the implementation from the scikit-learn tool [PVG<sup>+</sup>11] and all the algorithms were set to use the default parameter values. As regards BERT fine tuning, I used the stochastic gradient descent optimizer with a learning rate of 0.001, momentum 0.9 and ran the training process for 20 epochs. I set the batch size to 16 and limited the BERT sentence encoding to the maximum length of 80. In this work the BERT models were built based on the pytorch-pretrained-BERT repository<sup>6</sup>.

## 6.6 Results

I show the results in terms of macro average F-score. Table 6.4 shows the performance of the implemented models on all the datasets used in this work. The two best results obtained in each dataset is highlighted in bold face. Only the COVID and CRISISMMD datasets were split into training and validation sets by default, however to make it fair and comparable across all the datasets and approaches I performed 10-fold cross validation with the entire datasets (combined training and validation sets).

---

<sup>6</sup><https://github.com/huggingface/pytorch-pretrained-BERT>



In the first six rows I show the classification performance of conventional classifiers using the handcrafted features proposed in this work. For the full datasets it is only possible to calculate the Twitter-based features, as the user-based features are strongly dependent on the complete tweet information, and since I had to crawl the Twitter platform to obtain the complete information, I realised that many tweets had been deleted.

I noticed the performance of the classifiers varies on a per dataset basis and classifiers performed differently on each of the datasets. For the COVID dataset I observed the LOGISTIC REGRESSION classifier performed the best with Macro F1 of 57.07, while for CRISISLEXT6 datasets and CRISISLEXT26 MLP showed the best score 75.56 and 68.10, respectively. And for CRISISMMD, RANDOM FOREST outperformed the other classifiers with a score of 55.85.

The following six rows show the classification performance using Bag-of-Words as input features. Here again I noticed the performance of the classifiers varies on a per dataset basis, however I observed considerable performance improvement across all datasets which demonstrates that the bag-of-words is a stronger features encoding method than the handcrafted features approach only.

In the following six rows I show the results of the classification task using a combination of the handcrafted features with the Bag-of-Words features. It is interesting to observe that for the majority of the classifiers this combination does not produce improved results over the COVID and the CRISISLEXT6 datasets. Only NAIVE BAYES demonstrated considerable improvement over the previous approach for the COVID dataset. However, all the classifiers demonstrated improvement in the CRISISLEXT26 dataset when compared to using the Bag-of-Words only approach, and for the CRISISMMD dataset again only NAIVE BAYES demonstrated improvement when compared to the previous approach.

The next six rows show the results of the conventional classifiers using Fasttext word embeddings. For the COVID and CRISISLEXT6 datasets, MLP produced the best results, while for the CRISISLEXT26 and for the CRISISMMD datasets, LOGISTIC REGRESSION demonstrated the best macro F-score. In the following six rows I can see the classification results using GloVe word embeddings. The performance results observed from the classifiers using this embedding technique seem to be similar to the Fasttext word embeddings varying not too much across datasets.

In the following six rows I show the performance results of one approach in which I use the conventional classifiers using BERT encoded features combined with the handcrafted features. I have not noticed improvements using this approach of combining BERT word embeddings with handcrafted features on the COVID and CRISISMMD datasets, however I observed some improvements in the CRISISLEXT6 and CRISISLEXT26 dataset for the majority of the classifiers.

Finally in the last row I show the results of the proposed approach  $BERT_{Hyb}$ . The model outperforms all the previously cited methods across all datasets used in

this work. For COVID dataset it produced a macro F-score of 84.41 which is 2.5 percentage points improvement over the best result from previous approaches (LR using Bag-of-Words features). For CRISISLEXT6 I observed 95.96 macro F-score, for CRISISLEXT26 I obtained 79.09 macro F-score, which is the highest improvement (7 percentage points over SVM using handcrafted features combined with Bag-of-Words) and for CRISISMMD the model produced 77.66 macro F-score.

There are some reasons that can explain why the hybrid model performs much better than other models tested in this chapter. The first one is the fact that BERT encoder uses a contextual representation in which it processes words in relation to all the other words in the sequence, rather than one by one separately, and the second reason is the fact that I ran several training iterations while adjusting weights, and using different optimization functions to minimise the training loss.

Table 6.4: Models performance on the original datasets.

Features	Models	COVID	CRISISLEXT6	CRISISLEXT26	CRISISMMD
		MacroF1	MacroF1	MacroF1	MacroF1
HANDCRAFTED	LR	57.07(+/- 0.02)	75.09(+/- 0.14)	64.60(+/- 0.05)	48.91(+/- 0.02)
	DT	51.99(+/- 0.02)	72.39(+/- 0.14)	61.82(+/- 0.05)	55.63(+/- 0.02)
	RF	54.14(+/- 0.02)	74.05(+/- 0.14)	64.14(+/- 0.05)	55.85(+/- 0.02)
	NB	42.79(+/- 0.02)	72.51(+/- 0.14)	65.79(+/- 0.05)	50.60(+/- 0.02)
	MLP	49.84(+/- 0.02)	75.56(+/- 0.14)	68.10(+/- 0.05)	48.42(+/- 0.02)
	SVM	56.11(+/- 0.02)	75.41(+/- 0.14)	65.53(+/- 0.05)	49.81(+/- 0.02)
BAG-OF-WORDS	LR	<b>81.90</b> (+/- 0.04)	92.90(+/- 0.09)	66.46(+/- 0.17)	72.68(+/- 0.03)
	DT	75.13(+/- 0.04)	91.42(+/- 0.09)	53.04(+/- 0.17)	68.97(+/- 0.03)
	RF	81.06(+/- 0.04)	<b>93.51</b> (+/- 0.09)	62.59(+/- 0.17)	73.21(+/- 0.03)
	NB	66.75(+/- 0.04)	80.35(+/- 0.09)	57.09(+/- 0.17)	47.56(+/- 0.03)
	MLP	75.23(+/- 0.04)	91.74(+/- 0.09)	63.39(+/- 0.17)	71.48(+/- 0.03)
	SVM	81.38(+/- 0.04)	93.21(+/- 0.09)	65.01(+/- 0.17)	66.00(+/- 0.03)
HANDCRAFTED + BoW	LR	78.29(+/- 0.05)	83.58(+/- 0.12)	69.70(+/- 0.12)	65.12(+/- 0.03)
	DT	74.68(+/- 0.05)	90.80(+/- 0.12)	61.26(+/- 0.12)	66.24(+/- 0.03)
	RF	80.47(+/- 0.05)	93.28(+/- 0.12)	66.55(+/- 0.12)	70.61(+/- 0.03)
	NB	71.56(+/- 0.05)	79.00(+/- 0.12)	60.83(+/- 0.12)	57.53(+/- 0.03)
	MLP	75.28(+/- 0.05)	91.51(+/- 0.12)	63.96(+/- 0.12)	69.58(+/- 0.03)
	SVM	75.05(+/- 0.05)	92.96(+/- 0.12)	<b>72.09</b> (+/- 0.12)	66.18(+/- 0.03)
FASTTEXT	LR	77.60(+/- 0.04)	89.29(+/- 0.08)	71.30(+/- 0.10)	74.09(+/- 0.02)
	DT	64.42(+/- 0.04)	79.26(+/- 0.08)	60.85(+/- 0.10)	63.74(+/- 0.02)
	RF	76.16(+/- 0.04)	88.62(+/- 0.08)	69.83(+/- 0.10)	71.54(+/- 0.02)
	NB	74.73(+/- 0.04)	77.18(+/- 0.08)	63.41(+/- 0.10)	66.89(+/- 0.02)
	MLP	80.01(+/- 0.04)	91.28(+/- 0.08)	67.81(+/- 0.10)	74.00(+/- 0.02)
	SVM	76.43(+/- 0.04)	89.46(+/- 0.08)	70.14(+/- 0.10)	71.40(+/- 0.02)
GLOVE	LR	79.68(+/- 0.04)	86.82(+/- 0.11)	70.40(+/- 0.09)	74.59(+/- 0.02)
	DT	66.76(+/- 0.04)	77.49(+/- 0.11)	60.05(+/- 0.09)	63.54(+/- 0.02)
	RF	77.80(+/- 0.04)	87.36(+/- 0.11)	66.27(+/- 0.09)	72.41(+/- 0.02)
	NB	76.29(+/- 0.04)	81.72(+/- 0.11)	61.30(+/- 0.09)	72.87(+/- 0.02)
	MLP	79.03(+/- 0.04)	87.96(+/- 0.11)	66.21(+/- 0.09)	72.38(+/- 0.02)
	SVM	80.05(+/- 0.04)	89.20(+/- 0.11)	71.75(+/- 0.09)	<b>75.15</b> (+/- 0.02)
BERT	LR	77.83(+/- 0.03)	90.62(+/- 0.09)	70.41(+/- 0.10)	74.80(+/- 0.03)
	DT	62.19(+/- 0.03)	77.84(+/- 0.09)	60.98(+/- 0.10)	62.84(+/- 0.03)
	RF	74.11(+/- 0.03)	87.51(+/- 0.09)	69.11(+/- 0.10)	70.76(+/- 0.03)
	NB	71.34(+/- 0.03)	77.69(+/- 0.09)	67.59(+/- 0.10)	70.41(+/- 0.03)
	MLP	77.08(+/- 0.03)	89.75(+/- 0.09)	66.54(+/- 0.10)	72.21(+/- 0.03)
	SVM	78.08(+/- 0.03)	91.50(+/- 0.09)	70.53(+/- 0.10)	75.14(+/- 0.03)
HANDCRAFTED + BERT	BERT <sub>Hyb</sub>	<b>84.41</b> (+/- 0.01)	<b>95.96</b> (+/- 0.03)	<b>79.09</b> (+/- 0.04)	<b>77.66</b> (+/- 0.01)

I also evaluated the proposed approach in the subsets of the original datasets. As

mentioned before these subsets were created so I could also calculate features related to the user who posted the message. I noticed again that the handcrafted features alone did not produce satisfactory results. The best observed macro F-scores varied between 55.49 for the  $\text{CRISISMMD}_{\text{SUBSET}}$  using a NAIVE BAYES classifier and 78.58 for the  $\text{CRISISLEXt6}_{\text{SUBSET}}$  using RANDOM FOREST classifier. However when I used the Bag-of-Words model as input features, the classifiers produced considerably better results for  $\text{COVID}_{\text{SUBSET}}$  and  $\text{CRISISLEXt6}_{\text{SUBSET}}$  datasets in all cases, but for the  $\text{CRISISLEXt26}_{\text{SUBSET}}$  and  $\text{CRISISMMD}_{\text{SUBSET}}$  there were some classifiers that performed better using only the handcrafted features, for example for the  $\text{CRISISLEXt26}_{\text{SUBSET}}$  the RANDOM FOREST model produced a macro F-score of 66.80, while using the Bag-of-Words model it produced only 52.06. The combination of the handcrafted features and Bag-of-Words shows improvement for all datasets only when using the NAIVE BAYES classifier when compared to the Bag-of-Words model, while when compared to the sole handcrafted features the classifiers produce better results in all cases for the  $\text{COVID}_{\text{SUBSET}}$  and  $\text{CRISISLEXt6}_{\text{SUBSET}}$  datasets and the majority of cases in  $\text{CRISISLEXt26}_{\text{SUBSET}}$  and  $\text{CRISISMMD}_{\text{SUBSET}}$  with the exception of the NAIVE BAYES classifier.

Using the Fasttext, GloVe and BERT embeddings as input features to the conventional classifiers showed considerable improvements across all datasets, especially when using LOGISTIC REGRESSION as base classifier, however this was not a pattern observed when using different classification methods.

The hybrid model  $\text{BERT}_{\text{Hyb}}$  produced the best performance result for almost all the dataset with the exception of the  $\text{CRISISLEXt6}_{\text{SUBSET}}$ , however the difference is marginal. The best observed macro F-score is shown when using the Bag-of-Words features model using RANDOM FOREST as base classifier (93.22), while the hybrid approach produced a score of 93.05. In the  $\text{COVID}_{\text{SUBSET}}$  the model showed 84.64 macho F-score which is 2.3 percentage points improvement over the second best result (Bag-of-Words and LR = 82.35). The presented model showed 76.68 and 76.54 macro F-score for the  $\text{CRISISLEXt26}_{\text{SUBSET}}$  and  $\text{CRISISMMD}_{\text{SUBSET}}$  datasets respectively. These two datasets seem to be the two datasets where the performance of the models were lower than 80%. Further investigation and a more in depth analysis is required as there is still some room for improvements.

## 6.7 Conclusions and Future Works

Social media has drawn attention from different sectors of society and the information available during catastrophic events is extremely useful for both the ordinary citizen and the professionals involved in humanitarian purposes, however there is an overload of information that requires an automated filtering method for real time processing of relevant content.

In this work I designed a set of handcrafted features from both the Twitter posts

Table 6.5: Models performance on the subsets.

Features	Models	COVID <sub>SUBSET</sub>	CRISISLEXT6 <sub>SUBSET</sub>	CRISISLEXT26 <sub>SUBSET</sub>	CRISISMMD <sub>SUBSET</sub>
		MacroF1	MacroF1	MacroF1	MacroF1
HANDCRAFTED	LR	57.52(+/- 0.03)	76.56(+/- 0.12)	64.58(+/- 0.05)	48.44(+/- 0.03)
	DT	52.26(+/- 0.03)	71.04(+/- 0.12)	61.07(+/- 0.05)	54.00(+/- 0.03)
	RF	57.41(+/- 0.03)	78.58(+/- 0.12)	66.80(+/- 0.05)	54.29(+/- 0.03)
	NB	44.80(+/- 0.03)	73.40(+/- 0.12)	65.95(+/- 0.05)	55.49(+/- 0.03)
	MLP	50.47(+/- 0.03)	76.60(+/- 0.12)	67.03(+/- 0.05)	50.00(+/- 0.03)
	SVM	58.38(+/- 0.03)	77.09(+/- 0.12)	63.83(+/- 0.05)	48.66(+/- 0.03)
BAG OF WORDS	LR	<b>82.35</b> (+/- 0.03)	92.59(+/- 0.09)	67.52(+/- 0.14)	70.68(+/- 0.04)
	DT	74.36(+/- 0.03)	91.65(+/- 0.09)	52.06(+/- 0.14)	67.60(+/- 0.04)
	RF	81.55(+/- 0.03)	<b>93.22</b> (+/- 0.09)	63.66(+/- 0.14)	72.03(+/- 0.04)
	NB	67.24(+/- 0.03)	81.61(+/- 0.09)	58.73(+/- 0.14)	47.34(+/- 0.04)
	MLP	75.75(+/- 0.03)	91.21(+/- 0.09)	65.16(+/- 0.14)	69.76(+/- 0.04)
	SVM	81.24(+/- 0.03)	92.82(+/- 0.09)	65.18(+/- 0.14)	62.31(+/- 0.04)
HANDCRAFTED + BoW	LR	72.87(+/- 0.08)	81.16(+/- 0.14)	68.67(+/- 0.08)	62.25(+/- 0.03)
	DT	73.62(+/- 0.08)	90.58(+/- 0.14)	63.04(+/- 0.08)	64.84(+/- 0.03)
	RF	81.09(+/- 0.08)	92.64(+/- 0.14)	69.53(+/- 0.08)	67.10(+/- 0.03)
	NB	72.24(+/- 0.08)	82.48(+/- 0.14)	62.32(+/- 0.08)	55.25(+/- 0.03)
	MLP	75.99(+/- 0.08)	90.55(+/- 0.14)	65.90(+/- 0.08)	68.64(+/- 0.03)
	SVM	74.99(+/- 0.08)	92.31(+/- 0.14)	<b>72.99</b> (+/- 0.08)	63.50(+/- 0.03)
FASTTEXT	LR	78.11(+/- 0.04)	89.07(+/- 0.08)	70.78(+/- 0.09)	72.95(+/- 0.03)
	DT	64.73(+/- 0.04)	78.69(+/- 0.08)	61.32(+/- 0.09)	63.86(+/- 0.03)
	RF	76.05(+/- 0.04)	88.41(+/- 0.08)	68.84(+/- 0.09)	69.78(+/- 0.03)
	NB	75.18(+/- 0.04)	78.55(+/- 0.08)	62.17(+/- 0.09)	66.70(+/- 0.03)
	MLP	80.39(+/- 0.04)	91.53(+/- 0.08)	70.13(+/- 0.09)	73.70(+/- 0.03)
	SVM	76.88(+/- 0.04)	89.21(+/- 0.08)	68.11(+/- 0.09)	67.92(+/- 0.03)
GLOVE	LR	79.85(+/- 0.04)	86.77(+/- 0.10)	70.81(+/- 0.07)	74.35(+/- 0.03)
	DT	66.37(+/- 0.04)	76.47(+/- 0.10)	59.95(+/- 0.07)	63.62(+/- 0.03)
	RF	78.43(+/- 0.04)	87.21(+/- 0.10)	63.89(+/- 0.07)	71.55(+/- 0.03)
	NB	77.08(+/- 0.04)	82.16(+/- 0.10)	60.83(+/- 0.07)	72.44(+/- 0.03)
	MLP	79.40(+/- 0.04)	88.13(+/- 0.10)	68.52(+/- 0.07)	71.80(+/- 0.03)
	SVM	80.48(+/- 0.04)	89.09(+/- 0.10)	71.56(+/- 0.07)	<b>74.82</b> (+/- 0.03)
BERT	LR	78.08(+/- 0.03)	90.55(+/- 0.09)	70.44(+/- 0.09)	74.08(+/- 0.03)
	DT	62.71(+/- 0.03)	77.48(+/- 0.09)	61.34(+/- 0.09)	62.18(+/- 0.03)
	RF	74.55(+/- 0.03)	87.53(+/- 0.09)	67.96(+/- 0.09)	68.09(+/- 0.03)
	NB	71.89(+/- 0.03)	78.93(+/- 0.09)	67.07(+/- 0.09)	69.81(+/- 0.03)
	MLP	77.73(+/- 0.03)	89.59(+/- 0.09)	67.62(+/- 0.09)	71.33(+/- 0.03)
	SVM	78.47(+/- 0.03)	91.18(+/- 0.09)	71.42(+/- 0.09)	74.10(+/- 0.03)
HANDCRAFTED + BERT	BERT <sub>Hyb</sub>	<b>84.64</b> (+/- 0.01)	<b>93.05</b> (+/- 0.03)	<b>76.68</b> (+/- 0.04)	<b>76.54</b> (+/- 0.01)

and the users who posted a tweet, and showed experimentally the performance of six conventional classifiers on the informative tweet classification task. I also trained classifiers with several word embeddings, namely, Fasttext, GloVe and BERT, as input features. Moreover, I showed that the proposed deep neural model BERT<sub>Hyb</sub> is more effective in identifying informative tweets as compared to conventional classifiers in different crisis related corpus from Twitter.

As future works I intend to further investigate different deep learning models combinations and implement a complete pipeline where the tweets are crawled and classified in real time based on crisis related trending topics.

## **An Exploratory Analysis of Portuguese Tweets. Insights from Topics and Hashtags during Covid-19 Pandemic**

In this chapter I look at the conversation taking place on Twitter, with respect to Covid-19 in Brazil.

### **7.1 Introduction**

In December 2019 Chinese media channels reported that public health professionals were treating several cases of pneumonia of unknown causes in the city of Wuhan, the capital of Hubei, China. Shortly thereafter researchers identified a new virus that had infected dozens of people around the country with similar symptoms. However, only in January, Chinese state media reported to the general public the first known death caused by a virus of unknown etiology, which was followed by the announcement that researchers isolated this new virus from a sea food market in Wuhan.

Even though officials said they were monitoring it to prevent it from spreading, unfortunately, the interventions to contain an outbreak were not implemented soon enough and it rapidly spread to other countries. The spread of the new coronavirus, named as Covid-19, has created multiple hot-spots of the disease and the world has faced the worst pandemic event of the 21st century so far. Moreover the Covid-19 spread has put gigantic strains on many countries and economies around the world impacting not only health, but also the way we live and consume things.

In a recent study, Salgotra et al. [SGG20] modeled the effects of coronavirus in the fifteen most affected countries of the world. In Brazil, the first reported case was publicly confirmed in February in a traveler returning to São Paulo from northern Italy [JSC<sup>+</sup>20] and since then a rapid spread of the virus was observed.

Following the lead of other major countries, the Brazilian government has also

implemented social distancing measures, imposed ban on public gatherings and all other activities where there were chances of any social interactions.

The frequent use of social media has been extensively reported by previous researchers [EPS08, Lup14, Per15, KG19], but after governments imposed measures to contain the spread of Covid-19, the use of social media platforms has increased significantly and people started to use the social media platforms to shared their opinions and concerns about the pandemic more frequently [FBM<sup>+</sup>20]. Li et al. [LCC<sup>+</sup>20] showed a correlation of the Covid-19 related keywords search on social media with the incidence peak of the disease.

Generally speaking, social media has brought a huge change in people's behavior and preferences in recent years [Qua12] and Twitter is one of the most renowned social media platform that gets a huge amount of tweets every day serving as a valuable resource for communication [BL12, ZLCQ18], specially during the pandemic when it has facilitated understanding its impact, and most importantly established a link between public opinion and relevant actions and policies from both public and non-public organizations.

Although Twitter provides an excellent channel for public opinion creation and information sharing, it is a highly complex environment. Tweets are generally noisy, composed of incomplete and poorly structured sentences and irregular expressions, misspelled words and non-dictionary terms. Nonetheless the recent advances in artificial intelligence have contributed to establish an interest in both academia and industry as regards the development of studies for understanding the debate surrounding the Covid-19 pandemic through the social media perspective [IDKB20, MA21, AAKA<sup>+</sup>20].

In this chapter we look at the conversation taking place on social media, specifically Twitter, with respect to Covid-19 in Brazil. We describe a Portuguese Twitter dataset with more than 4 million tweets collected during a 16-month period. We search for insights with descriptive textual analytics and data visualization, such as exploratory Word Clouds. We investigate popular keywords shared among Twitter users. We also apply topic modeling and sentiment analysis methods to investigate questions related to the topics evolution over time as well as the sentiment expressed by users on Twitter during the pandemic.

This dataset is useful for researchers who want to conduct comparative and analytical studies on the perception of the pandemic on social networks. The dataset was carefully collected since the beginning of the outbreak in December 2019 until March 2021. Therefore, it is a temporal dataset, which aggregates additional value to the corpus.

The remaining of the chapter is organized as follows. We briefly introduce some related works in the next section, followed by a description of our methods for collecting, parsing and cleaning the raw tweets. Then we describe an experimental setup in which we first analyse the content of the corpus, followed by presenting results from

experiments on topic modeling, hashtags suggestion and sentiment analysis. Finally in the last section we draw some conclusions and point out future directions.

## 7.2 Related Works

Social media channels are among the most worldwide used means of communication on the internet. Twitter for example plays an important role in the reporting and dissemination of news events where users share opinions and short messages.

In the context of health, social media research was primarily focused on examining the patient experience. Alemiet et al. [ATCA12] for instance, performed sentiment analysis on patient's comments from different sources to assist designing real-time satisfaction surveys.

Afyouni et al. [AFA15] investigated users' opinions through sentiment analysis by calculating in-degree centralities of nodes to identify the hubs in the network of interactions and observed that the overall opinion about digital healthcare is usually positive. Moreover, Benetoli et al. [BCA18] studied how the use of social media impacted patients interaction with healthcare professionals.

For a more complete study on the effects of social media use by patients, the work by Smailhodzic et al. [SHBL16] provides interesting observations by performing extensive literature review of the utilization of social media in healthcare.

Twitter has been successfully employed as an important venue for mining, tracking and forecasting previous health crises [OY15, MJL<sup>+</sup>19, PDB14, AGL<sup>+</sup>12]. Twitter content offers the advantage of being freely available in real-time. During the Covid-19 pandemic, Twitter has been widely used to capture self reported symptoms [SLHB<sup>+</sup>20] as well as to explore fake news and rumors related to the pandemic itself [AVADS20]. Moreover other researchers have investigated the sentiment dynamics on Twitter conversations regarding the Covid-19 topic [GZJ<sup>+</sup>20, KvdVM20, SLB<sup>+</sup>20] and to assess mitigation strategies such as social distancing [YFR<sup>+</sup>20, KGFF20].

In summary, the amount of conversation that takes place about a certain topic on Twitter gives us insights into the amount of attention that is placed on a certain topic and lately we have observed an increase in the amount of messages posted on Twitter relating to the Covid-19 pandemic.

## 7.3 Methods

We describe the steps performed to execute our analysis setup and experimental settings.

### 7.3.1 Data Acquisition

We adopt an approach based on content analysis, aiming to observe and analyse how the users communicate in social networks since the beginning of the Covid-19 pandemic outbreak. This kind of analysis helps us identifying and categorizing representation groups, understanding and interpreting the current reality.

Users constantly post tweets that can be sorted into categories by the inclusion of hashtags, keywords, or phrases beginning with a hash mark (#) and ending in white space, within the bodies of the tweets. Besides enabling the users to directly access data by using specific tags, they are also used to express messages to specific groups and highlight relevant happenings. The increase of Covid-19 related hashtags use during the pandemic is an important indicator of the importance of the subject among users and the level of information as regards the disease.

In order to create a corpus of Covid-19 related tweets, we collected the raw tweets from the Twitter platform using its API <sup>1</sup> and considered a set of seed keywords in Portuguese related to the pandemic. We manually created a list of hashtags from empirical observations of different media channels and news agencies from Brazil. The initial list of seed terms can be seen on Table 7.1. We used this list of 50 hashtags and performed a first pass by fetching tweets posted between December 2019 and March 2021. Moreover, we collected only the Portuguese tweets by using the language parameter and setting it to 'pt' value. Shortly thereafter, we performed a second pass in which we identified the top-10 most used hashtags for each month and used this list of terms to re-fetch more tweets. It is important to mention that we manually excluded recurring hashtags from the initial list of seed terms in these month-wise list of top-10 hashtags. This step would guarantee we would not collect repeated tweets. Moreover, in case of repetition of tweets due to the occurrence of another hashtag, we also removed it from the tweets collection.

### 7.3.2 Preprocessing

Tweets are usually short and full of informal language. Thus, the first step in any text processing task is to pre-process and clean the data. In order to so, we follow a series of steps that are applied in a given order to improve the text. Below we describe the steps we followed.

- Most of the social media platforms use hashtags to identify key topics and keywords relating to a certain event, for example *#Coronavirus*, *#Pandemia*, *#Fiqueemcasa*, etc., thus we performed basic cleaning of the text by removing the hashtag character and kept the hashtag term.
- We removed hyperlinks, digits, punctuation, symbols as well as special characters and discarded the tweet if it was left empty after this step.

---

<sup>1</sup><https://developer.twitter.com/en/docs/api-reference-index>



Table 7.1: List of seed hashtags

#auxilioemergencial	#azitromicina	#aztrazeneca	#cloroquina
#coronavac	#coronavirus	#coronavirusnobrasil	#coronavirusbrasil
#covidnobrasil	#covidbrasil	#covid	#covid19
#covid19brasil	#combateaocorona	#distanciamentosocial	#ficaemcasa
#fiocruz	#fiqueemcasa	#hidroxicloroquina	#ivermectina
#isolamentosocial	#lockdown	#novocoronavirus2019	#novocoronavirus
#novocoronavirus2020	#oxfordastrazeneca	#pandemia	#pfizer
#quarentena	#saicorona	#saicovid	#saudepublica
#sinovac	#sputnik	#vacinacovid	#vacinacoronavac
#vacinadachina	#vacinadarussia	#vacinadeoxford	#vacinadobutantan
#vacinaja	#vacinapfizer	#vacinasinovac	#vacinasputnik
#vemvacina	#viruscorona	#viruscovid	#viruscovid19
#virusdachina	#viruschines		

- Twitter uses the @ symbol to prefix usernames and allows a user to tag other users. For this case we removed the complete text including the @ symbol and the account name of the tagged user.
- To avoid recognizing the same word (i.e. *#COVID* and *#covid*) due to the capitalization we fold all capitalized letter to lower case.
- Considering we are analysing Portuguese text we created a list of Portuguese stop-words to remove them from the tweets. Removing stop words is a very popular method to reduce the noise in text by removing words that occur very frequently but are not informative, for example, articles, prepositions and adverbs.
- Occasionally, the Twitter API returns duplicate tweets. Therefore we compared the tweets and discarded duplicates in order to avoid putting extra weight on any particular tweet.

### 7.3.3 Text Representation

Text representation is a fundamental problem in Information Retrieval, in which its main goal is to obtain a numerical representation of any text for machines to understand.

In traditional Information Retrieval the first techniques proposed were considered discrete text representation models, for example One-Hot encoding, Bag-of-words (BOW), Advanced BOW - TF-IDF, among others. In these kinds of techniques, the words are represented by their corresponding indexes to their position in a dictionary from a larger corpus. There are some disadvantages in these techniques such as the high dimensionality which can be computationally expensive, these representations

cannot produce co-occurrence statistics between words, i.e they assume all words are independent of each other and the positional information of the words cannot be captured either.

Distributed representation of words have been used in several NLP related tasks as an improvement to the discrete text representation models. One famous implementation of distributed representation is the Word2Vec model from Mikolov et al. [MCCD13]. It is a group of unsupervised algorithms for creating word embeddings from text documents. To train word embeddings, Word2Vec uses a two layer neural network to process unlabeled documents. The neural network architecture is based either on the continuous bag of words (CBOW) or the skip-gram. In the CBOW approach the input to the model for a word,  $w_i$  are the words preceding and succeeding this word, i.e.,  $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$  when using two words before and after the current word. The output of the network is the probability of  $w_i$  being the correct word. The task can be described as predicting a word given its context. Moreover, in the skip-gram model the input to the model is a word  $w_i$  and the Word2Vec model predicts the surrounding context words  $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$ .

### 7.3.4 Deep Transfer Learning

Deep learning has significantly improved the state of the art for several machine learning applications, such as speech recognition [DHK13], computer vision [PW17], text understanding [MPGC17], among others. It enables computational models composed of multiple layers to learn representations of data with multiple levels of abstraction.

Several deep learning models have been developed over the years, for example Stacked Auto-Encoder (SAE), Deep Belief Network (DBN), Deep Generative Networks (DGN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNNs), Recursive Neural Network (RvNN), among others.

BERT [DCLT19] is an advanced pre-trained language representation model that makes use of an attention mechanism that learns contextual relations between words (or sub-words) in a text. It uses a bidirectional Transformer network to pre-train a language model on a large corpus and enables a fine-tuning approach of the model's parameters on other similar tasks. In a classification task, the first token of a sequence is identified with a unique token [CLS] and a fully-connected layer is connected at that token's position of the last encoder layer, finally a softmax layer completes the sentence or the sentence-pair classification.

Lately, Transfer Learning has been widely studied to overcome challenges such as the laborious manual annotation of large corpora and the expensive cost of retraining deep neural models in every single new target domain. Transfer learning aims to build learning machines that generalize across different domains following different probability distributions [SNK<sup>+</sup>07, PTKY10, DTX12, ZSMW13]. It can be done by using the pre-trained networks as fixed feature extractors or by fine-tuning the weights of the pre-trained models.

## 7.4 Experiments and Analysis

In this section we present our experiments. Firstly we describe the corpus and show some statistics about the corpus of tweets we created as well as we describe three experiments we perform on the collected tweets.

### 7.4.1 Corpus analysis

Initially we take a look at the content of the conversations taking place on Twitter by investigating the most prevalent words relating to Covid-19.

Following the line of thought from the previous sections we collected more than 4 million tweets within the time period of December 2019 and March 2021. In Table 7.2 we show the tweets distribution on a monthly basis while in Table 7.3 we show the number of hashtags along the months.

Even though Asian and European countries had already registered hundreds of cases of Covid-19 contamination, in Brazil the first contamination was only identified in the end of February <sup>2</sup> and the first death was officially registered only in March. During the months of March and April in 2020 we observed a considerable increase in the number of tweets as well as the number of retweets. This behavior is in accordance to the first wave of Covid-19 contamination in Brazil. Moreover, starting from mid January 2021 we faced the second wave of contamination with the P.1 variant. Figure 7.1 shows the number of tweets and retweets overtime and it is easy to observe the high peaks specially during these two specific waves of contamination.

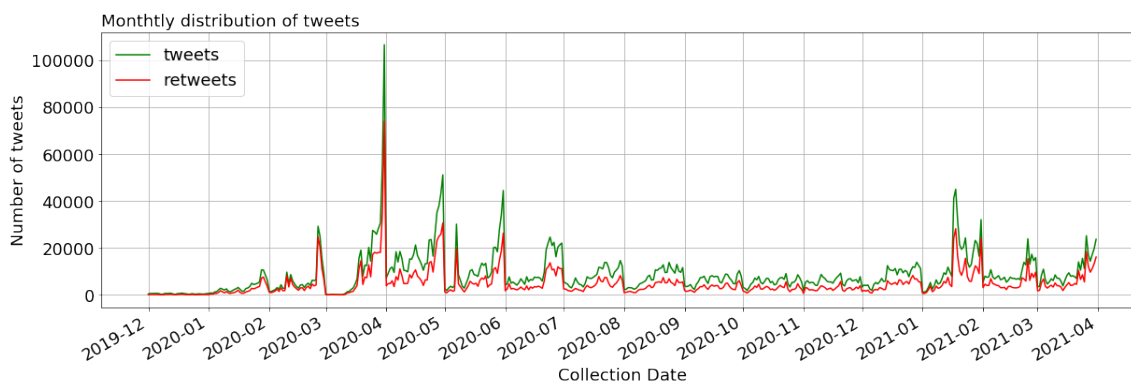


Figure 7.1: Monthly distribution of tweets and retweets

Moreover, we observe that April is the month with the highest number of tweets and hashtags. This fact is in accordance to when the Brazilian government adopted social distancing measures and submitted a proposal for emergency financial support to being voted in Congress.

<sup>2</sup><https://www.medscape.com/viewarticle/925806>



Figure 7.2: Most mentioned Twitter accounts

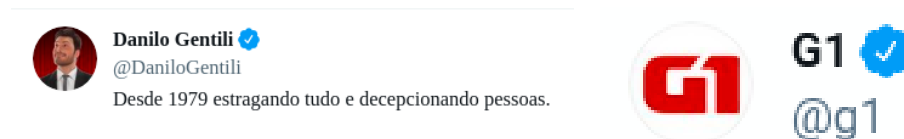


Figure 7.3: Most influential user accounts

Politicians, media channels and health organizations are the most frequent user mentions, with the Brazilian president *@jairbolsonaro* being by far the most frequent user mention in the collected corpus with more than 400 thousand occurrences followed by one of the major television channel’s account *@g1* and the official Ministry of Health’s account *@minsaude* (Figure 7.2). After manually inspecting the tweets that were mentioning the current president’s account during the pandemic period it is easy to observe that the population in general was desperately claiming for some assistance as well as demonstrating dissatisfaction with the current political situation.

The hashtag *#coronavirus* was by far the most cited one followed by *#covid19* and *#ficaemcasa*. Considering the hashtag *#coronavirus* is the most common term used in other languages as well, we have observed other variations with a high frequency, for example *#coronavírus*, *#covid19brasil* and *#coronavirusnobrasil* referring to a similar and central concept which is the pandemic event. As regards the public claim for people to stay at home and adopt the social distancing measures, we have seen variations such as *#fiqueemcasa*, *#ficaemcasa*, *#ficoemcasa*, among others. Moreover, when it comes to the topic related to the vaccination the most relevant hashtags were *#vacinaja*, *#vacina* and *#vacinaparatodos*.

### 7.4.2 Popular domains

Identifying the URLs shared by Twitter users can tell us a lot about the quality of the information shared within users about a certain topic. We used a regular expression to extract all the URLs strings from the tweets. Since they are mostly shortened URLs, we then expanded and examined these shared links to determine the most popular domains.

Figure 7.4 presents the top-10 domains with respect to their frequency in tweets having more than 100 user accounts citing that domain. Twitter itself, turned out to

Table 7.2: Month-wise tweets distribution

PERIOD	# tweets
Dec/19	16,584
Jan/20	106,112
Feb/20	186,809
Mar/20	427,419
Apr/20	555,795
May/20	369,878
Jun/20	320,639
Jul/20	251,962
Aug/20	239,809
Sep/20	190,966
Oct/20	176,559
Nov/20	180,591
Dec/20	255,273
Jan/21	445,691
Feb/21	266,881
Mar/21	313,242
<b>TOTAL</b>	<b>4,304,210</b>

Table 7.3: Month-wise hashtags distribution

PERIOD	#hashtags
Dec/19	78,988
Jan/20	316,216
Feb/20	359,001
Mar/20	865,909
Apr/20	1,255,631
May/20	890,023
Jun/20	861,038
Jul/20	792,074
Aug/20	732,426
Sep/20	613,150
Oct/20	572,900
Nov/20	586,804
Dec/20	659,823
Jan/21	980,429
Feb/21	683,376
Mar/21	684,423
<b>TOTAL</b>	<b>10,932,211</b>

be by far the most linked URL, followed by Instagram. Moreover the remaining links mainly belong to news outlets and government agencies.

There is no easy way to verify the credibility of the information shared by users considering only the domain, rather than the content of the messages, however this methodology has been used before for similar purposes [SCV<sup>+</sup>18, BB20]. For a more recent study on the quality of the URLs shared on Twitter, refer to the article by Singh et al. [SBB<sup>+</sup>20].

### 7.4.3 Words being used

In this section, we investigate the content of the conversations taking place about Covid-19.

One important aspect of textual analytics involves the identification of the most frequent terms. Thus, we begin by looking at the most frequent n-gram in each tweet.

Figure 7.5 shows the top 10 unigram, bigram and trigram, excluding stop-words.

Not surprisingly the term *covid* was the most used unigram with more than one hundred thousand uses followed by *brasil* and *vacina*. The most frequent bigram (two words sequence) observed is *contra covid*, followed by *primeira dose* and *vacina brasileira* which indicates an increase on the usage of terms more related to the vac-

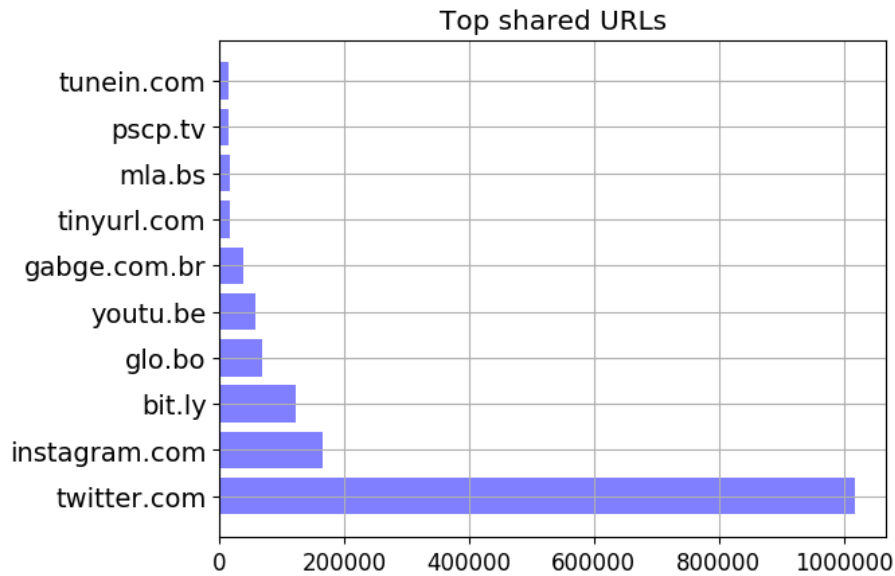


Figure 7.4: Top 10 most popular domains

ination itself. Moreover this consistency continues when we look at the top trigrams (three words sequences by the use of *vacina brasileira contra*, *brasileira contra covid* and *butantanvac vacina brasileira*, and a direct mention to the brasilian vaccination against the virus.

We have also explored longer sequences but the corpus did not contain longer sequences with sufficient frequency relevance. To expand on this analysis we created word clouds for each month. Word clouds provide a visual representation of text appearing in tweets by highlighting words according to frequency of appearance.

We demonstrate on Figure 7.6 some word clouds created from the texts in our collected datasets. The words are sized based on how frequently they appear in the corpora with larger words appearing in more tweets. It is notorious that in the first word cloud the vocabulary was more general while in the later months the word clouds appear to be more specific to the pandemic event with a focus on the nature of the virus and words describing the virus and its spread.

#### 7.4.4 Topic Modeling

Topic modeling is a technique used to discover and summarize the main topic from a corpus of documents automatically. Among the existing techniques, one of the most common is the Latent Dirichlet Allocation (LDA), which represents each topic as a probability distribution over the words in a dictionary. Godin et al. [GSDN+13], for instance, employed LDA to model the underlying topic assignment of tweets, however LDA suffers a large performance degradation over short texts.

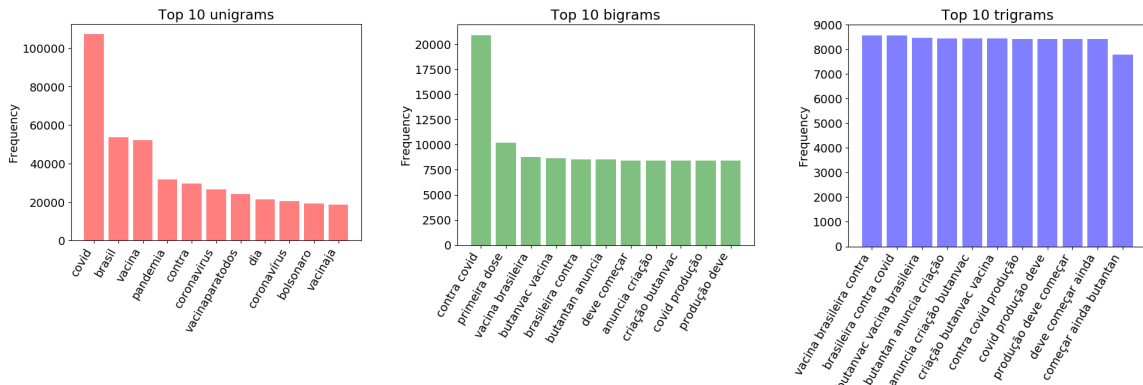
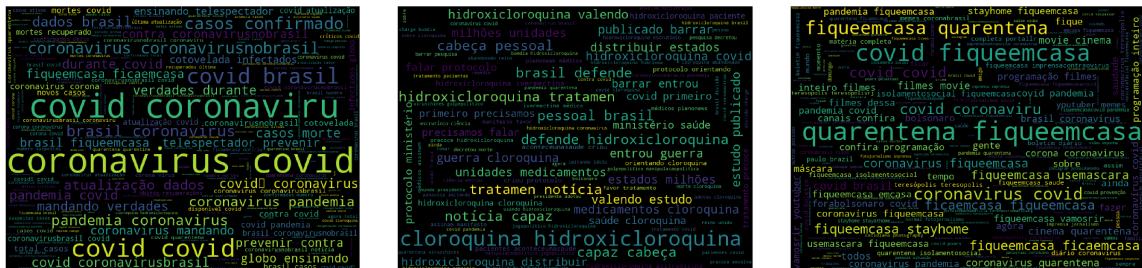


Figure 7.5: Top 10 n-grams

When it comes to Twitter posts, hashtags are generally used to summarize the tweet's contents as well as serving as proxy to help to categorizing and searching for tweets. Moreover, in the scientific literature, a topic is often defined by a single hashtag. Though this may be a too restrictive choice for many cases if one consider that a single hashtag is loaded with meanings and it is used as a means to express one's opinion.

We extend the definition of a topic to be more embracing. We start considering a seed hashtag and we define a topic as being a set of  $n$  related hashtags that co-occur with the seed hashtag. We label this topic and consequently this topic is used to get all the tweets that contain at least one of the hashtags that co-occur with the seed hashtag. For example the seed term *#hidroxicloroquina* co-occurred with the hashtag *#coronavirus* in 2286 documents, with the hashtag *#Brasil* in 1502 documents, with the hashtag *#ivermectina* in 371 documents and so on. Figure 7.6 shows the word clouds for three topics defined and labeled as previously described. The word clouds refer to the topics *#coronavirus*, *#hidroxicloroquina* and *#fiqueemcasa* respectively.

Figure 7.6: Word clouds for the topics *coronavirus*, *hidroxicloroquina* and *fiqueemcasa* respectively

### 7.4.5 Hashtags Suggestion

Hashtags are useful in several natural language tasks, for example tweets classification, searching and clustering, social network analysis, among others.

Manually searching for hashtags that are relevant to a certain concept is a laborious task, therefore, an automatic method for discovering a relevant hashtag can be useful in a variety of scenarios, for example it can help users to reach popularity trends and drive up fast engagement, it can help users to pick the hashtags that works best for some marketing campaign, it can be used for query suggestions and increase recall of a query, etc.

Khabiri[KCK12] has proposed a content-based hashtag recommendation method. In their method, they take into consideration the content of a tweet to recommend hashtags based on the content of a tweet, where the tweet is represented by a bag of words and the relevance between a word and a hashtag is measured on a hashtag-word co-occurrence graph.

Similar to the work proposed by Li et al. [LSF<sup>+</sup>16] we define a list of hashtags and then we calculate the top 10 hashtags that are closest to each initial hashtag in terms of cosine similarity according to a word embedding. A word embedding is a dense, high-dimensional and real-valued vector representation that encodes the meaning of words such that similar words are closer in this high-dimensional space. One of the most famous word embeddings method is the *Word2Vec* which has been used in several natural language processing applications and has been proposed in the work by Mikolov et al. [MCCD13].

We benefit from a *Word2Vec* model previously trained with a large Portuguese corpora of mixed documents<sup>3</sup> and we extend the training process with our corpus of tweets.

Finally we obtain a hashtag embedding by querying the trained model and calculating the cosine similarity score between this hashtag's embedding vector and the embedding vector of each hashtag in the model. Then, the list of top 10 hashtags with the highest score is selected. Table 7.4 shows one example (the hashtag *#vacina*) and the top 10 most similar hashtags as well as their respective similarity score.

By visually inspecting this table it is clear that the top 10 hashtags are indeed related to the queried hashtag *#vacina*, for example the first one (i.e *#oxford*) was the most used hashtag to refer to the vaccine produced by Oxford.

### 7.4.6 Sentiment Analysis

Sentiment analysis is the process of identifying people's sentiment based on some texts they produced, for example a Twitter message. It has become very popular in research due to the vast amount of opinionated texts produced by Internet users on social

---

<sup>3</sup><http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>



Table 7.4: Most similar hashtags to the hashtag *#vacina*

<b>#hashtags</b>	<b>similarity</b>
#oxford	0.589
#jacaré	0.566
#aprovada	0.558
#butanta	0.556
#vacine	0.554
#sinovac	0.549
#vacinas	0.546
#butantan	0.546
#doses	0.545

networks. Moreover, analyzing the sentiment on Twitter can help understanding how people feel about some specific topic or event.

Dealing with micro-posts can be a challenging task which requires an extra effort since well-established techniques do not perform well on this type of data. Gimpel et al. [GSO<sup>+</sup>10], for example, shows that in this context, problems like tokenization and POS tagging, are much more difficult to deal with than in normal texts. Moreover, labeled data for training a model is not always available at hand. Thus transfer learning approaches can address this problem by exploiting a pre-trained model and a labeled source domain as starting point and to obtain a model for a target domain that is different from the source domain.

We applied transfer learning methodology to calculate the sentiment observed on tweets posted during the Covid-19 pandemic in Brazil.

We employed a BERT model for the Portuguese language [SNL20] that is pre-trained on the brWaC (Brazilian Web as Corpus), a large open source Brazilian Portuguese language corpus. Also, we benefit from huggingface<sup>4</sup> pytorch-transformer implementation [WDS<sup>+</sup>20] that includes a set of interfaces designed for a series of NLP tasks and we use *BertForSequenceClassification* model. We fine tune this model using a data set of Portuguese tweets available on Kaggle<sup>5</sup> platform. This corpus is made of four parts, being two of them tweets classified as positive and negative and with tweets about politics in one of them and no specific theme in the other, plus two other files with tweets extracted from news channels which are closer to a neutral sentiment. We chose to discard the neutral sentiment tweets as we wanted to treat it as a binary classification problem. Moreover we discarded the file related to politics theme, since we constructed a corpus of Covid-19 related tweets that is not related to politics. This left us with a dataset of of 785,814 tweets.

<sup>4</sup><https://huggingface.co/transformers/>

<sup>5</sup><https://www.kaggle.com/augustop/portuguese-tweets-for-sentiment-analysis>

## Baseline methods

We created some baseline methods to compare with our proposed method. The first one was the Bag-Of-Words model weighted by TF-IDF, the second one was the WORD2VEC and the third one was the GLOVE model. Both the WORD2VEC and GLOVE models were obtained from a repository of pre-trained models<sup>6</sup>. We used the Skip-gram model with 300 dimensions and only extended the models by training them with our corpus of tweets.

We encoded the input corpus of tweets and used them as input features with the following classifiers: NAIVE BAYES - a simple and effective model that is often used for text classification that is based on the Bayes theorem [Lew92], LOGISTIC REGRESSION - a popular method for classification in which the probabilities describing the possible outcomes of a single trial are modeled using a logistic function [PVG+11], DECISION TREE - a tree-based model of decisions in which any path beginning from the root is described by a series of basic tests and RANDOM FOREST - consisting of a large number of individual decision trees where each individual tree spits out a class prediction and the class with the most votes is considered as the final model's prediction.

## Evaluation Metrics

To compare the performance of the classification methods, we look at a set of standard performance measures. Accuracy (Acc) is the most basic classification evaluation measure and is computed as  $Acc = \frac{TP+TN}{N}$  where N is the total number of the testing instances, TP is the True Positive and TN is the True negatives. Precision (P) and Recall (R) are given as follows:  $P = \frac{TP}{TP+FP}$  and  $R = \frac{TP}{TP+FN}$ . F-measure is the harmonic meant of both Precision and Recall and is given as follows:  $F = \frac{2*Precision*Recall}{Precision+Recall}$ .

Table 7.5 shows the results for the sentiment prediction task in terms of Macro Accuracy, Precision, Recall and F-score, respectively.

In the first four rows we can see the LOGISTIC REGRESSION model outperformed all the other classifiers in terms of Macro F-score(71.05) when using the Bag-of-Words as input features. The following four rows of the table shows the classification performace when using the WORD2VEC embeddings as input features. We observed the best classifier was the RANDOM FOREST followed by the LOGISTIC REGRESSION model. They produced a Macro F-score of 70.29 and 70.08 respectively.

The next four rows shows the classifiers performance when using GLOVE embeddings and the best performance was observed when using the RANDOM FOREST classifier with a Macro F-score of 63.10. For the majority of cases the classifiers produced worse results when using the WORD2VEC and GLOVE than by using the traditional Bag-Of-Words technique. An explanation to that may be the fact that in

---

<sup>6</sup><http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>

Table 7.5: Macro Acc,P,R,F

	clf.	Acc	P	R	F
TF-IDF	LR	75.44	73.68	70.29	71.05
	DT	70.54	67.24	65.89	66.27
	RF	74.56	72.41	69.49	70.17
	NB	69.34	66.52	65.53	65.42
WORD2VEC	LR	75.39	72.94	69.01	70.08
	DT	68.95	64.86	64.39	64.59
	RF	76.25	74.83	68.98	70.29
	NB	59.79	62.27	63.52	59.40
GLOVE	LR	71.14	67.78	62.03	62.47
	DT	65.70	60.94	60.44	60.63
	RF	72.74	71.64	62.70	63.10
	NB	56.60	59.05	59.97	56.20
BERT		79.43	77.36	75.13	75.99

Table 7.6: BERT fine tuned model per class prediction

	P	R	F
POSITIVE	0.82	0.88	0.85
NEGATIVE	0.72	0.62	0.67

both cases the models were pre-trained on larger corpus of long documents and suffers degradation when being evaluated on short and noisy texts. Finally in the last row we show our approach of transfer learning using the BERT transformer. Our method shows superior performance in terms of Macro F-score (75.99) when compared to all other methods tested.

In Table 7.6 we show per class prediction results. When it comes to predicting the positive classes the models performs better and produces a Macro F-score of 0.85 while for the negative classes the same model produces a Macro F-score of 0.67. We observed the number of positive classified tweets were double the number of negative classified tweets, and in general the positive tweets were referring to the financial support offered by the federal government, the vaccination production as well as acknowledgments to professionals involved in saving lives. While for the negative classified tweets, they were mostly mentioning number of contamination and deaths. Further analysis is still required to understand the low performance on classifying negative labeled tweets.

Limitations of this study relies on the fact that it was performed on a single

language (Portuguese) and considering Twitter is a highly dynamic environment there is the possibility that tweets may be excluded due to numerous reasons, and thus one may not be able to reproduce exactly the same results. Nevertheless we made available all the models, algorithms and the corpus produced in this study<sup>7</sup>.

## 7.5 Conclusions

Social media platforms have become extremely popular among users worldwide. And the analysis of social media can help us understand better the society and how things evolve. The current pandemic has changed a lot the way people live and how they behave on the social media platforms. Understanding how people behave and how they express their opinions and sentiments on the internet are extremely important.

In this chapter we have explored Twitter and analysed 16 months of tweets posted by users in Brazil referring to the Covid-19 pandemic. We have performed several experiments over the collected corpus, such as topic identification based on hashtags co-occurrences, hashtag recommendation using distributed representation of words and sentiment analysis with a deep transfer learning approach.

Our findings showed the volume of messages correlates with the external events such as the number of Covid-19 cases, the vaccines, financial support being offered to the population, among others.

Future works include improving the performance of the neural model as well as a deeper understanding of the miss-classified messages. Furthermore, since this study was limited to a single language and only Twitter, we plan to extend it to more languages as well as investigate other social networks.

---

<sup>7</sup>Anonymous link

## Conclusions and Future Works

### 8.1 Conclusions

In this chapter, I draw main conclusions from findings presented in this thesis. In the first part of the thesis I have focused in the study of NLP techniques towards EL approaches. In Chapter 2 I have investigated a novel problem of detecting and understanding EL difficulty by introducing a method to generate difficulty labels for entity mentions in arbitrary corpora. My approach to detect difficult to link mentions can improve the performance of state-of-the-art EL tools, by enabling the efficient prediction of critical cases which require manual labelling. To this end I have introduced a set of features that can be used within a distantly supervised model for predicting difficult to link mentions on the fly, for cases where no labels can be assigned by the proposed labelling method or when real time analysis is needed.

In Chapter 3 I have studied the temporal variability of entity mentions and why it should be taken into account for entity linking system's evaluations. I conducted experiments with different Wikipedia editions and also created an entity linking model that uses the entity's prior probability calculated over different Wikipedia snapshots.

In Chapter 4 I have analysed the performance of different EL systems and showed the performance of EL systems can be optimised by combining the results of distinct EL systems in an ensemble fashion. In this study I introduced a novel approach called Meta Entity Linking modeled as a supervised classification task for predicting the EL system that can provide the correct link for an ambiguous mention.

In the second part of this thesis I have focused mainly on social media platforms, specially Twitter. In Chapter 6 I proposed a deep neural model BERTHyb to identify informative tweets during catastrophic events. I proposed a set of handcrafted features from both the Twitter posts and the users who posted the message, and showed the performance of six conventional classifiers in comparison to my BERTHyb approach. My approach showed to be more effective as compared to conventional classifiers in different crisis related corpus from Twitter.

In Chapter 7 I have explored Twitter and analysed 16 months of tweets posted by users in portuguese referring to the Covid-19 pandemic. I have performed several experiments over the collected corpus, such as topic identification based on hashtags co-occurrences, hashtag recommendation using distributed representation of words and sentiment analysis with a deep transfer learning approach. The experiments showed the volume of messages correlates with the external events such as the number of Covid-19 cases, the vaccines, financial support being offered to the population, among others.

## 8.2 Future Works

Building on observations and findings presented in this thesis, I plan to investigate the following aspects in the future. As regards the entity linking tasks as well as the prediction of difficult to link entity mentions I am concerned with studying more informative features that can reflect in better prediction of difficult to link mentions, specially for the minority class (HARD). Moreover I plan to evaluate the performance gain of the MetaEL approach using different combinations of EL systems as well as investigate whether I can benefit from the use of more advanced models for the binary classification task, in order to improve its (relatively low) performance.

As regards the works presented in this thesis studying social media, at first I intend to investigate different deep learning models combinations and implement a complete pipeline where the tweets are crawled and classified in real time based on crisis related trending topics. Followed by a deeper investigation of how to improve the performance of the proposed model as well as a deeper understanding of the misclassified messages. In the study I conducted about the messages posted by users during the pandemic, I plan to extend it in more depth to other languages since it was limited to a single language.

## Bibliography

- [AAKA<sup>+</sup>20] Raghad Alshalan, Hend Al-Khalifa, Duaa Alsaeed, Heyam Al-Baity, and Shahad Alshalan. Detection of hate speech in covid-19–related tweets in the arab region: Deep learning and topic modeling approach. *Journal of Medical Internet Research*, 22(12):e22609, 2020.
- [AFA15] Soroosh Afyouni, Ahmed E Fetit, and Theodoros N Arvanitis. # digitalhealth: Exploring users’ perspectives through social media analysis. In *Enabling Health Informatics Applications*, pages 243–246. IOS Press, 2015.
- [AGL<sup>+</sup>12] Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Twitter improves seasonal influenza prediction. In *Healthinf*, pages 61–70, 2012.
- [AHH<sup>+</sup>12] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Semantics+ filtering+ search= twitcident. exploring information in social web streams. In *HT’12*, 2012.
- [AOI18] Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *ICWSM*, 2018.
- [AR17] Flavia Sofia Acerbo and Claudio Rossi. Filtering informative tweets during emergencies: a machine learning approach. In *Proceedings of the First CoNEXT Workshop on ICT Tools for Emergency Networks and DisastEr Relief*, 2017.
- [ATCA12] Farrokh Alemi, Manabu Torii, Laura Clementz, and David C Aron. Feasibility of real-time satisfaction surveys through automated analysis of patients’ unstructured comments and sentiments. *Quality Management in Healthcare*, 21(1):9–19, 2012.

- [AVADS20] Wasim Ahmed, Josep Vidal-Alaball, Joseph Downing, and Francesc López Seguí. Covid-19 and the 5g conspiracy theory: social network analysis of twitter data. *Journal of medical internet research*, 22(5):e19458, 2020.
- [AZ12] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.
- [BB07] Nguyen Bach and Sameer Badaskar. A survey on relation extraction. *Language Technologies Institute, Carnegie Mellon University*, 178:15, 2007.
- [BB20] Lia Bozarth and Ceren Budak. Toward a better performance evaluation framework for fake news classification. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 60–71, 2020.
- [BBDM15] Abdelghani Bouziane, Djelloul Bouchiha, Nouredine Doumi, and Mimioun Malki. Question answering systems: survey and trends. *Procedia Computer Science*, 73:366–375, 2015.
- [BCA18] A Benetoli, TF Chen, and P Aslani. How patients’ use of social media impacts their interactions with healthcare professionals. *Patient education and counseling*, 101(3):439–444, 2018.
- [BDR17] K. Bontcheva, L. Derczynski, and I. Roberts. Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*. Springer, 2017.
- [BEP<sup>+</sup>08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [BGJM16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- [BGM14] Dan Brickley, Ramanathan V Guha, and Brian McBride. Rdf schema 1.1. *W3C recommendation*, 25:2004–2014, 2014.
- [BL12] Axel Bruns and Yuxian Eugene Liang. Tools and methods for capturing twitter data during natural disasters. *First Monday*, 2012.
- [BP06] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Eacl*, volume 6, pages 9–16, 2006.



- [CANG16] F. Corcoglioniti, A. P. Aprosio, Y. Nechaev, and C. Giuliano. Microneel: Combining nlp tools to perform named entity detection and linking on microposts. In *CLiC-it/EVALITA*, 2016.
- [CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
- [CHLL18] Y. Cao, L. Hou, J. Li, and Z. Liu. Neural collective entity linking. In *COLING*, 2018.
- [CJY<sup>+</sup>16] L. Chen, J. M. Jose, H. Yu, F. Yuan, and H. Zhang. Probabilistic topic modelling with semantic graph. In *ECIR*. Springer, 2016.
- [CLT18] L. Canale, P. Lisena, and R. Troncy. A novel ensemble method for named entity recognition and disambiguation based on neural network. In *ISWC*. Springer, 2018.
- [CMP11] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *WWW*, 2011.
- [CPPR<sup>+</sup>16] A. E. Cano, D. Preotiuc-Pietro, D. Radovanović, K. Weller, and A. Dadzie. #microposts2016: 6th workshop on making sense of microposts: Big things come in small packages. In *WWW. IW3C2*, 2016.
- [CPR12] Mark A Cameron, Robert Power, Bella Robinson, and Jie Yin. Emergency situation awareness from twitter for crisis management. In *WWW*, 2012.
- [CST16] Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. Identifying informative messages in disaster events using convolutional neural networks. In *ISCRAM*, 2016.
- [Cuc07] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.
- [CXQ15] Gong Cheng, Danyun Xu, and Yuzhong Qu. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *Proceedings of the 24th International Conference on World Wide Web*, pages 184–194, 2015.
- [DBD<sup>+</sup>14] Elena Demidova, Nicola Barbieri, Stefan Dietze, Adam Funk, Helge Holzmann, Diana Maynard, Nikolaos Papailiou, Wim Peters, Thomas Risse, and Dimitris Spiliotopoulos. Analysing and enriching focused

- semantic web archives for parliament applications. *Future Internet*, 6(3):433–456, 2014.
- [DCLT19] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- [DDC12] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 469–478, 2012.
- [DHK13] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8599–8603. IEEE, 2013.
- [DKC<sup>+</sup>14] S. Dlugolinskỳ, P. Krammer, M. Ciglan, M. Laclavík, and L. Hluchỳ. Combining named entity recognition methods for concept extraction in microposts. *Making Sense of Microposts (# Microposts2014)*, 2014.
- [DMM<sup>+</sup>10] Meghan Dougherty, Eric T Meyer, Christine McCarthy Madsen, Charles Van den Heuvel, Arthur Thomas, and Sally Wyatt. Researcher engagement with web archives: State of the art. *Joint Information Systems Committee Report*, 2010.
- [DMP<sup>+</sup>04] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC, Lisbon, Portugal*, volume 2, page 1, 2004.
- [DP02] James Price Dillard and Michael Pfau. *The persuasion handbook: Developments in theory and practice*. Sage Publications, 2002.
- [DTX12] Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.
- [Ela05] Pradheep Elango. Coreference resolution: A survey. *University of Wisconsin, Madison, WI*, page 12, 2005.
- [EPS08] Nina Eyrich, Monica L Padman, and Kaye D Sweetser. Pr practitioners’ use of social media tools and communication technology. *Public relations review*, 34(4):412–414, 2008.

- [F<sup>+</sup>98] Christiane Fellbaum et al. Wordnet: An electronic lexical database mit press. *Cambridge, Massachusetts*, 1998.
- [FBM<sup>+</sup>20] Blossom Fernandes, Urmi Nanda Biswas, Roseann Tan Mansukhani, Alma Vallejo Casarín, and Cecilia A Essau. The impact of covid-19 lockdown on internet use and escapism in adolescents. *Revista de Psicología Clínica con Niños y Adolescentes*, 7(3):59–65, 2020.
- [FC14] Yuan Fang and Ming-Wei Chang. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, 2, 2014.
- [FCL<sup>+</sup>19] Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. Joint entity linking with deep reinforcement learning. In *The world wide web conference*, pages 438–447, 2019.
- [FHKN17] Pavlos Fafalios, Helge Holzmann, Vaibhav Kasturia, and Wolfgang Nejdl. Building and querying semantic layers for web archives. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–10. IEEE, 2017.
- [FIJZ03] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *NAACL*. ACL, 2003.
- [Fis97] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *ASRU*. IEEE, 1997.
- [FKS03] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. In *ACM-SIAM, January 12-14, 2003, Baltimore, Maryland, USA.*, pages 28–36, 2003.
- [FMN18] Zeon Trevor Fernando, Ivana Marenzi, and Wolfgang Nejdl. Archiveweb: collaboratively extending and exploring web archive collections—how would you like to work with your collections? *International Journal on Digital Libraries*, 19(1):39–55, 2018.
- [FS10] Paolo Ferragina and Ugo Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628, 2010.
- [FSEC09] Anthony Fader, Stephen Soderland, Oren Etzioni, and Turing Center. Scaling wikipedia-based named entity disambiguation to arbitrary web text. In *IJCAI Workshop on User-contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, 2009.

- [GC14] Daniel Gomes and Miguel Costa. The importance of web archives for humanities. *International Journal of Humanities and Arts Computing*, 8(1):106–123, 2014.
- [GH17] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*, 2017.
- [GHB<sup>+</sup>13] D. Gerber, S. Hellmann, L. Bühmann, T. Soru, R. Usbeck, and A. N. Ngomo. Real-time rdf extraction from unstructured data streams. In *ISWC*. Springer, 2013.
- [GNMC09] Daniel Gomes, André Nogueira, João Miranda, and Miguel Costa. Introducing the portuguese web archive initiative. In *8th international Web archiving workshop*. Springer, 2009.
- [Gra] David Graff. *The aquaint corpus of English news text:[content copyright] Portions© 1998-2000 New York Times, Inc.,© 1998-2000 Associated Press, Inc.,© 1996-2000 Xinhua News Service*. Linguistic Data Consortium.
- [GRS<sup>+</sup>18] David Graf, Werner Retschitzegger, Wieland Schwinger, Birgit Pröll, and Elisabeth Kapsammer. Cross-domain informativeness classification for disaster situations. In *MEDES*, 2018.
- [GSDN<sup>+</sup>13] Frédéric Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd international conference on world wide web*, pages 593–596, 2013.
- [GSO<sup>+</sup>10] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2010.
- [GZJ<sup>+</sup>20] Junling Gao, Pinpin Zheng, Yingnan Jia, Hao Chen, Yimeng Mao, Suhong Chen, Yi Wang, Hua Fu, and Junming Dai. Mental health problems and social media exposure during covid-19 outbreak. *Plos one*, 15(4):e0231924, 2020.
- [HFH<sup>+</sup>09] Mark A. Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

- [HGT09] Wei Che Darren Huang, Shlomo Geva, and Andrew Trotman. Overview of the *inex 2009 link the wiki track*. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 312–323. Springer, 2009.
- [HHJ15] Hongzhao Huang, Larry Heck, and Heng Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*, 2015.
- [HLL<sup>+</sup>13] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, 2013.
- [HNA17] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. Exploring web archives through temporal anchor texts. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 289–298. ACM, 2017.
- [HNF<sup>+</sup>16] Benjamin D. Horne, Dorit Nevo, Jesse Freitas, Heng Ji, and Sibel Adali. Expertise in social networks: How do experts differ from other users? In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 583–586, 2016.
- [HRN<sup>+</sup>13] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.
- [HSB<sup>+</sup>11] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. YAGO2: Exploring and querying world knowledge in time, space, context and many languages. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 229–232, 2011.
- [HSN<sup>+</sup>12] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *CIKM*. ACM, 2012.
- [HYB<sup>+</sup>11] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Conference on Empirical Methods in Natural Language Processing*, 2011.
- [HZ09] Xianpei Han and Jun Zhao. Nlpr.kbp in tac 2009 kbp track: A two-stage method to entity linking. *Theory and Applications of Categories*, 2009.

- [ICDV15] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *CSUR*, 2015.
- [ICL<sup>+</sup>14] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. Aidr: Artificial intelligence for disaster response. In *WWW*, 2014.
- [IDKB20] Ali Shariq Imran, Sher Muhammad Daudpota, Zenun Kastrati, and Rakhi Batra. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets. *IEEEr Access*, 8:181074–181090, 2020.
- [IEC<sup>+</sup>13] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Extracting information nuggets from disaster-related messages in social media. In *Iscram*, 2013.
- [JFD19] Renato Stoffalette João, Pavlos Fafalios, and Stefan Dietze. Same but different: distant supervision for predicting and understanding entity linking difficulty. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1019–1026, 2019.
- [JFD20] Renato Stoffalette João, Pavlos Fafalios, and Stefan Dietze. Better together: an ensemble learner for combining the results of ready-made entity linking systems. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 851–858, 2020.
- [JLMR16] Andrew Jackson, Jimmy Lin, Ian Milligan, and Nick Ruest. Desiderata for exploratory search interfaces to web archives in support of scholarly activities. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 103–106. IEEE, 2016.
- [JSC<sup>+</sup>20] Jaqueline Goes de Jesus, Claudio Sacchi, Darlan da Silva Candido, Ingra Morales Claro, Flávia Cristina Silva Sales, Erika Regina Manuli, Daniela Bernardes Borges da Silva, Terezinha Maria de Paiva, Margarete Aparecida Benega Pinho, Katia Correa de Oliveira Santos, et al. Importation and early local transmission of covid-19 in brazil, 2020. *Revista do Instituto de Medicina Tropical de Sao Paulo*, 62, 2020.
- [KBR20] Marc-André Kaufhold, Markus Bayer, and Christian Reuter. Rapid relevance classification of social media posts in disasters and emergencies: A system and evaluation featuring active, incremental and online learning. *IPM*, 57(1), 2020.
- [KCK12] Elham Khabiri, James Caverlee, and Krishna Y Kamath. Predicting semantic annotations on the real-time web. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 219–228, 2012.

- [KG19] Kagan Kircaburun and Mark D Griffiths. Problematic instagram use: The role of perceived feeling of presence and escapism. *International Journal of Mental Health and Addiction*, 17(4):909–921, 2019.
- [KGFF20] Jiye Kwon, Connor Grady, Josemari T Feliciano, and Samah J Fodeh. Defining facets of social distancing during the covid-19 pandemic: twitter analysis. *Journal of biomedical informatics*, 111:103601, 2020.
- [KGH18] N. Kolitsas, O. Ganea, and T. Hofmann. End-to-end neural entity linking. In *CoNLL*. ACL, 2018.
- [KJMH<sup>+</sup>19] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.
- [KM11] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.
- [KSRC09] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *SIGKDD, Paris, France, June 28 - July 1, 2009*, pages 457–466, 2009.
- [KvdVM20] Bennett Kleinberg, Isabelle van der Vegt, and Maximilian Mozes. Measuring emotions in the covid-19 real world worry dataset. *arXiv preprint arXiv:2004.04225*, 2020.
- [LCC<sup>+</sup>20] Cuilian Li, Li Jia Chen, Xueyu Chen, Mingzhi Zhang, Chi Pui Pang, and Haoyu Chen. Retrospective analysis of the possibility of predicting the covid-19 outbreak from internet searches and social media data, china, 2020. *Eurosurveillance*, 25(10):2000199, 2020.
- [LCCH18] Hongmin Li, Doina Caragea, Cornelia Caragea, and Nic Herndon. Disaster response aided by tweet classification with a domain adaptation approach. *JCCM*, 2018.
- [Lew92] David Dolan Lewis. *Representation and learning in information retrieval*. PhD thesis, University of Massachusetts Amherst, 1992.
- [LIJ<sup>+</sup>15] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

- [LSF<sup>+</sup>16] Quanzhi Li, Sameena Shah, Rui Fang, Armineh Nourbakhsh, and Xiaomo Liu. Discovering relevant hashtags for health concepts: A case study of twitter. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [LSHL20] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.
- [LT18] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. *arXiv preprint arXiv:1804.10637*, 2018.
- [LT19] Phong Le and Ivan Titov. Distant learning for entity linking with automatic noise detection. *arXiv preprint arXiv:1905.07189*, 2019.
- [Lup14] Deborah Lupton. ‘feeling better connected’: Academics’ use of social media. 2014.
- [MA21] SreeJagadeesh Malla and PJA Alphonse. Covid-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets. *Applied Soft Computing*, page 107495, 2021.
- [Mar12] Angel R Martinez. Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):107–113, 2012.
- [MBB<sup>+</sup>17] Jose G Moreno, Romaric Besançon, Romain Beaumont, Eva D’hondt, Anne-Laure Ligozat, Sophie Rosset, Xavier Tannier, and Brigitte Grau. Combining word and entity embeddings for entity linking. In *European Semantic Web Conference*, pages 337–352. Springer, 2017.
- [MC07] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
- [MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [Mei13] Patrick Meier. Crisis maps: Harnessing the power of big data to deliver humanitarian assistance. *Forbes Magazine*, 2013.
- [MHK14] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.



- [MJGSB11] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8, 2011.
- [MJL<sup>+</sup>19] Shahir Masri, Jianfeng Jia, Chen Li, Guofa Zhou, Ming-Chieh Lee, Guiyun Yan, and Jun Wu. Use of twitter data to improve zika virus surveillance in the united states during the 2016 epidemic. *BMC public health*, 19(1):1–14, 2019.
- [MPGC17] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.
- [MRN14] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.
- [MSC<sup>+</sup>13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [MUC95] *MUC6 '95: Proceedings of the 6th Conference on Message Understanding*, USA, 1995. Association for Computational Linguistics.
- [MW08] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008.
- [NAMJ<sup>+</sup>17] Dat Tien Nguyen, Kamla Al-Mannai, Shafiq R Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. *ICWSM*, 2017.
- [Nav09] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.
- [NCC18] Venkata Kishore Neppalli, Cornelia Caragea, and Doina Caragea. Deep neural networks versus naive bayes classifiers for identifying informative tweets during disasters. In *ISCRAM*, 2018.
- [NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.

- [NVR<sup>+</sup>20] Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*, 2020.
- [NXC<sup>+</sup>16] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu, and S. S. Cao. Semantic documents relatedness using concept graph representation. In *WSDM*. ACM, 2016.
- [OCDV14] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *ICWSM*, 2014.
- [OVC15] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *CSCW*, 2015.
- [OY15] Michelle Odlum and Sunmoo Yoon. What can we learn about the ebola outbreak from tweets? *American journal of infection control*, 43(6):563–571, 2015.
- [PDB14] Michael J Paul, Mark Dredze, and David Broniatowski. Twitter improves influenza forecasting. *PLoS currents*, 6, 2014.
- [Per15] Andrew Perrin. Social media usage. *Pew research center*, 125:52–68, 2015.
- [PF14] Francesco Piccinno and Paolo Ferragina. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62. ACM, 2014.
- [PRT16] J. Plu, G. Rizzo, and R. Troncy. Enhancing entity linking by combining ner models. In *Semantic Web Evaluation Challenge*. Springer, 2016.
- [Pru08] Eric Prud’hommeaux. Sparql query language for rdf, w3c recommendation. <http://www.w3.org/TR/rdf-sparql-query/>, 2008.
- [PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [PTKY10] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

- [PVG<sup>+</sup>11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 2011.
- [PW17] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
- [Qua12] Erik Qualman. *Socialnomics: How social media transforms the way we live and do business*. John Wiley & Sons, 2012.
- [RKC16] H. Raviv, O. Kurland, and D. Carmel. Document retrieval using entity-based language models. In *SIGIR*. ACM, 2016.
- [RP15] P. Ruiz and T. Poibeau. Combining open source annotators for entity linking through weighted voting. In *\*SEM*, 2015.
- [RRDA11] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1375–1384, 2011.
- [RRPH16] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes. MEKA: A multi-label/multi-target extension to Weka. *Journal of Machine Learning Research*, 2016.
- [RT11] G. Rizzo and R. Troncy. Nerd: A framework for evaluating named entity recognition tools in the web of data. In *ISWC*, 2011.
- [RTV18] Priya Radhakrishnan, Partha Talukdar, and Vasudeva Varma. Elden: Improved entity linking using densified knowledge graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1844–1853, 2018.
- [RUNN17] M. Röder, R. Usbeck, and A. Ngonga Ngomo. Gerbil—benchmarking named entity recognition and linking consistently. *Semantic Web*, 2017.
- [RVT<sup>+</sup>13] Jakob Rogstadius, Maja Vukovic, Claudio A Teixeira, Vassilis Kostakos, Evangelos Karapanos, and Jim Alain Laredo. Crisistracker: Crowdsourced social media curation for disaster awareness. *IBM Journal of Research and Development*, 2013.
- [San08] Evan Sandhaus. The New York Times Annotated Corpus LDC2008T19. *Philadelphia: Linguistic Data Consortium*, 2008.

- [SBB<sup>+</sup>20] Lisa Singh, Leticia Bode, Ceren Budak, Kornraphop Kawintiranon, Colton Padden, and Emily Vraga. Understanding high-and low-quality url sharing on covid-19 twitter streams. *Journal of computational social science*, 3(2):343–366, 2020.
- [SC12] Valentin I Spitzkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. 2012.
- [SCV<sup>+</sup>18] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature communications*, 9(1):1–9, 2018.
- [SDM03] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
- [SE13] S. Saha and A. Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, 2013.
- [SGG20] Rohit Salgotra, Mostafa Gandomi, and Amir H Gandomi. Evolutionary modelling of the covid-19 pandemic in fifteen most affected countries. *Chaos, Solitons & Fractals*, 140:110118, 2020.
- [SHA16] Jaspreet Singh, Johannes Hoffart, and Avishek Anand. Discovering entities with just a little help from you. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1331–1340, 2016.
- [SHBL16] Edin Smailhodzic, Wyanda Hooijsma, Albert Boonstra, and David J Langley. Social media use in healthcare: A systematic review of effects on patients and on their relationship with healthcare professionals. *BMC health services research*, 16(1):1–14, 2016.
- [SHK<sup>+</sup>14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.
- [SJ17] Renato Stoffalette João. Time-aware entity linking. In *Doctoral Consortium at ISWC, Austria*, 2017.
- [SJ20] Renato Stoffalette João. On the temporality of priors in entity linking. In *European Conference on Information Retrieval*, pages 375–382. Springer, 2020.

- [SJ21a] Renato Stoffalette João. On informative tweet identification for tracking mass events. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*,, pages 1266–1273. INSTICC, SciTePress, 2021.
- [SJ21b] Renato Stoffalette João. A semantic layer querying tool. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1101–1104, 2021.
- [SKH05] Luo Si, Tapas Kanungo, and Xiangji Huang. Boosting performance of bio-entity recognition by combining results from multiple systems. In *Proceedings of the 5th international workshop on Bioinformatics*, pages 76–83. ACM, 2005.
- [SKW07] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.
- [SLB<sup>+</sup>20] Leonard Schild, Chen Ling, Jeremy Blackburn, Gianluca Stringhini, Yang Zhang, and Savvas Zannettou. ” go eat a bat, chang! ”: An early look on the emergence of sinophobic behavior on web communities in the face of covid-19. *arXiv preprint arXiv:2004.04046*, 2020.
- [SLHB<sup>+</sup>20] Abeed Sarker, Sahithi Lakamana, Whitney Hogg-Bremer, Angel Xie, Mohammed Ali Al-Garadi, and Yuan-Chi Yang. Self-reported covid-19 symptoms on twitter: an analysis and a research resource. *Journal of the American Medical Informatics Association*, 27(8):1310–1315, 2020.
- [SLT<sup>+</sup>15] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.
- [SN14] R Speck and A. N. Ngomo. Ensemble learning for named entity recognition. In *ISWC*. Springer, 2014.
- [SNK<sup>+</sup>07] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, volume 7, pages 1433–1440. Citeseer, 2007.
- [SNL20] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. BERTimbau: pre-trained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.

- [SSA<sup>+</sup>20] Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*, 2020.
- [SWH14] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2014.
- [TK07] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *IJDWM*, 2007.
- [UNR<sup>+</sup>14] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In *ISWC 2014, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 457–471, 2014.
- [URN15] R. Usbeck, M. Röder, and A. N. Ngonga. Evaluating entity annotators using gerbil. In *ISWC*. Springer, 2015.
- [VHSP10] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *SIGCHI*, 2010.
- [VK14] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
- [VPK<sup>+</sup>09] Vasudeva Varma, Prasad Pingali, Rahul Katragadda, Sai Krishna, Surya Ganesh, Kiran Sarvabhotla, Harish Garapati, Hareen Gopisetty, Vijay Bharath Reddy, Kranthi Reddy, et al. Iiit hyderabad at tac 2009. In *TAC*, 2009.
- [VSP<sup>+</sup>17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.
- [VVC<sup>+</sup>11] Sudha Verma, Sarah Vieweg, William J Corvey, Leysia Palen, James H Martin, Martha Palmer, Aaron Schram, and Kenneth Mark Anderson. Natural language processing to the rescue? extracting” situational awareness” tweets during mass emergency. In *ICWSM*, 2011.
- [WDS<sup>+</sup>20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger,

- Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [WNC03] D. Wu, G. Ngai, and M. Carpuat. A stacked, voted, stacked model for named entity recognition. In *NAACL. ACL*, 2003.
- [YB19] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.
- [YFR<sup>+</sup>20] Joseph Younis, Harvy Freitag, Jeremy S Ruthberg, Jonathan P Romanes, Craig Nielsen, and Neil Mehta. Social media as an early proxy for social distancing indicated by the covid-19 reproduction number: Observational study. *JMIR Public Health and Surveillance*, 6(4):e21340, 2020.
- [YIR18] Yi Yang, Ozan Irsoy, and Kazi Shefaet Rahman. Collective entity disambiguation with structured gradient tree boosting. *arXiv preprint arXiv:1802.10229*, 2018.
- [YSTT16] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*, 2016.
- [ZLCQ18] Lei Zou, Nina SN Lam, Heng Cai, and Yi Qiang. Mining twitter data for improved understanding of disaster resilience. *Annals of the American Association of Geographers*, 108(5):1422–1441, 2018.
- [ZSG16] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and Collective Entity Disambiguation through Semantic Embeddings. In *SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 425–434, 2016.
- [ZSMW13] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013.
- [ZWL18] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.