UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

BSI
Sistemas de Informação
UFRPE

Alesson Delmiro Francisco

Aspect Term Extraction in Aspect-Based Sentiment Analysis

Recife, Brazil

December - 2019

Alesson Delmiro Francisco

Aspect Term Extraction in Aspect-Based Sentiment Analysis

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Supervisor: Rinaldo José de Lima

Recife, Brazil

December - 2019

ALESSON DELMIRO FRANCISCO

ASPECT TERM EXTRACTION IN ASPECT-BASED SENTIMENT ANALYSIS

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 22 de Dezembro de 2019.

BANCA EXAMINADORA

Rinaldo José de Lima (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Silvana Bocanegra
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Vanilson André de Arruda Burégio
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

F819a Francisco, Alesson Delmiro
Aspect term extraction in aspect-based sentiment analysis / Alesson Delmiro Francisco. - 2019.
66 f. : il.

Orientador: Rinaldo José de Lima.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, 2019.

1. Opinion target extraction. 2. CRF. 3. Aspect-Based Sentiment Analysis. I. Lima, Rinaldo José de, orient. II. Título

CDD 004

Acknowledgements

My special thanks go to my wife, Manoela Cirilo, for all the support she gave every time I needed and all the incentive to carry on in my weakest moments.

Special thanks also to my mother, Luciana Delmiro, always supported me throughout the time in graduation and gave me the upbringing that got me into university and made me the person I am today.

I would like to thank my supervisor, Rinaldo Lima, for the opportunity to work this project, his help and orientation, which was primordial to the completion of this work.

It is the quality of one's convictions that determines success, not the number of followers.
- Remus J. Lupin, "Harry Potter and the Deathly Hallows" by J.K. Rowling

Resumo

O uso crescente da Internet criou a necessidade de analisar uma vasta quantidade de dados. Uma grande quantia de dados é apresentada como Texto em Linguagem Natural não estruturado, com várias maneiras de expressar a mesma informação. É uma tarefa importante extrair informação e significado destes conteúdos não estruturados, como opiniões em produtos ou serviços. A necessidade de extrair e analisar a vasta quantidade de dados criados todos os dias na Internet ultrapassou as capacidades humanas, como resultado, várias aplicações de mineração de texto que extraem e analisam dados textuais produzidos por humanos estão disponíveis atualmente, uma destas aplicações é a Análise de Sentimentos usada para que empresas e provedores de serviços possam usar o conhecimento extraído de documentos textuais para melhor entender como seus clientes pensam sobre eles. No entanto, a tarefa de analisar texto não estruturado é difícil, por isso é necessário prover informação coerente e resumos concisos para as revisões. Análise de Sentimento é o processo de identificar e categorizar computacionalmente opiniões expressadas num texto, especialmente para determinar a atitude do autor sobre um tópico ou produto em particular. Análise de Sentimentos Baseada em Aspectos (ABSA) é um sub-campo da Análise de Sentimentos que tem como objetivo extrair opiniões mais refinadas e exatas, quebrando o texto em aspectos. A maior parte dos trabalhos atuais na literatura não lucram de recursos baseados em semântica ou análises baseadas em Processamento de Linguagem Natural na fase de pré-processamento. Para tratar essas limitações, um estudo nestes recursos é feito com o objetivo de extrair as características necessárias para a execução da tarefa, e para fazer a melhor combinação para Extração de Termo de Aspecto. Este trabalho tem como o principal objetivo implementar e analisar um método de Extração de Termo de Aspecto (ATE) de críticas de usuários (restaurantes e laptops). O método proposto é baseado em uma abordagem supervisionada chamada Campos Condicionais Aleatórios (CRF) que otimiza o uso de características para classificação, esta escolha é justificada pelos trabalhos relacionados anteriores que demonstram a eficácia do CRF para ATE. Um estudo também é feito em métodos para propor novas características e experimantar com combinações de características para obter as melhores combinações. O estudo detalhado é feito a partir da experimentação com características de palavra, n-gramas e características customizadas utilizando um algoritmo supervisionado CRF para realizar a tarefa de Extração de Termo de Aspecto com resultados em termo de Precisão, Cobertura e F-Measure, as métricas padrões de avaliação adotadas na área. Por fim, uma avaliação comparativa entre o método proposto para ATE contra outros trabalhos da literatura mostra que o método apresentado neste trabalho é competitivo.

Palavras-chave: Extração de alvo de Opinião, CRF, Análise de Sentimentos Baseada em Aspectos.

Abstract

The increasing use of the Internet in many directions has created a necessity to analyze a large quantity of data. A large amount of data is presented as Natural Language Text, which is unstructured, with many ways to express the same information. It is an important task to extract information and meaning from those unstructured content, such as opinions on products or services. The need to extract and analyze the large amount of data created every day on the Internet surpassed the capabilities of human ability, as a result, many text mining applications that extract and analyze textual data produced by humans are available today, one of such kind of applications is Sentiment Analysis, viewed as a vital task both to the academic and commercial fields, so that companies and service providers can use that knowledge extracted from textual documents to better understand how their customers think about them or to know how their products and services are appreciated or not by their customers. However, the task of analysing unstructured text is a difficult one, that is why it is necessary to provide coherent information and concise summaries to those revisions. Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine the writer's attitude towards a particular topic or product. Aspect-Based Sentiment Analysis is a sub-field of Sentiment Analysis that aims to extract more refined and exact opinions, by breaking down text into aspects. Most of the current work in the literature does not take profit of either semantic-based resources or NLP-based analysis in the preprocessing stage. To countermeasure these limitations, a study on these resources is done aiming to extract the features needed to execute the task, and to make the best combination for ATE. This work has the main goal of implementing and analysing a method of Aspect Term Extraction (ATE) of users reviews (restaurants and laptops). The proposed method is based on a supervised approach called Conditional Random Fields (CRF) which is able to optimize the use of features for classification, this choice was justified by previous related work that demonstrate the effectiveness of CRF for ATE. Also, we are investigating the existing methods and features for ABSA, as well as proposing new features and experimenting with feature combinations in order to find the best features combinations, that are not yet covered in the state of art. The detailed study is done by experimenting with word features, n-grams and custom made features using an CRF supervised algorithm to accomplish the task of Aspect Term Extraction with results in terms of Precision, Recall and F-measure, the standard evaluation metrics adopted in the field. Finally, a comparative assessment between the proposal method for ATE against other related work presented in the literature has shown that the method presented by this work is competitive.

Keywords: Opinion target extraction, CRF, Aspect-Based Sentiment Analysis.

List of Figures

List of Tables

List of abbreviations and acronyms

ABSA	Aspect-Based Sentiment Analysis
ACD	Aspect Category Detection
ATE	Aspect Term Extraction
CRF	Conditional Random Fields
IE	Information Extraction
IR	Information Retrieval
MT	Machine Translation
NLP	Natural Language Processing
PoS	Part of Speech
RNN	Recurrent Neural Network
SPC	Sentiment Polarity Classification

Contents

1 Introduction

In recent years, there has been an enormous growth in Internet use and online interactions such as social media, chats, and forums which creates a large number of data, mostly presented in form of natural language texts, which are inherently unstructured. Internet was also changed by the way the users behave online, e.g., instead of being sole consumers, internet users have become content creators. Among the large spectrum of content produced by Internet users, there exists an important piece of information being created every day: opinions.

Users have the power to popularize or criticize a product or a service with a simple review on the web. This relevant piece of data has been used by many companies in order to produce more refined products or services to their potential customers. The process of systematic feedback has also been important to academic studies, with a large spectrum of works, dedicated to the study of opinions, called Sentiment Analysis.

Sentiment Analysis (SA) or Opinion Mining is the field of study dedicated to the study of people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes (LIU, 2012). Opinion mining is a part of text mining that focus on the processing of user generated content, that are as mentioned before, unstructured and can be about the most diverse subjects in the form o free text. This is a characteristic that adds difficulty to research challenges such as topic and opinion identification.

A sub-field of Sentiment Analysis of particular interest is Aspect-Based Sentiment Analysis (ABSA), that proposes approaches with many potential applications to be explored, with demonstrations of good results in the literature. ABSA, being a specific sub-task of Sentiment Analysis, produces a more refined approach to SA because it is a technique that breaks down text into aspects (attributes or components of a product or service), and then allocates each one a sentiment level (positive, negative or neutral), it is used to analyze different features/attributes/aspects of a product. In Aspect-Based Sentiment Analysis, an aspect of an opinion is the target term in which the opinion is referred to.

Although there have been many advances on the field, Sentiment Analysis still has a large number of unsolved or partially solved challenges. One challenge for SA is the natural language processing overhead such as co-reference, inference, ambiguity and inference. Another challenge can be contextual information or the use of metaphors, such information can be difficult to automatically obtain from sources like social media websites; Words orientation problem represents yet another challenge for SA researches, since the same word can be of different polarity when used in different contexts. Other examples of challenges are the use of abbreviation or short words and opinions and reviews in different languages. These factors combined with domain, text type and text level make the task of establishing the state of art for opinion mining a laborious one. In addition, there is a demand for more precise and accurate methods to produce more innovative approaches to improve the approaches based on aspects. Due to this context, this research aims to analyze Aspect Term Extraction.

The motivation for this work is the growing demand for tools to process opinions, both in the academic field and corporation, where companies need to evaluate opinions on their products. In addition, given the broadness of the field, the possibility of improving the current state of art in the literature in the ways of most optimal feature combination to yield better results for Aspect Term Extraction in Sentiment Analysis. The current approaches still do not present a satisfactory method to extract aspect terms using a CRF supervised algorithm. With this context, this works aims to answer the question of what is the combination of features that results in the best Precision, Recall and F-measure for the task of Aspect Term Extraction in a CRF supervised algorithm in annotated sentences.

The scope of this work is addressing the task of Aspect Term Extraction by evaluating and experimenting with combination of features to define the most optimal of these combinations to extract aspect terms in restaurants and laptops reviews with annotated sentences. In this case, the ABSA sub-task of Aspect Category Detection is not addressed in this work, as it was the initial aim to combine these two sub-tasks. Also the remaining subtasks of ABSA were not addressed: Aspect Term Polarity and Aspect Category Polarity.

1.1 Goals

1.1.1 Main Goal

The main research goal is to propose, implement, and evaluate a supervised method for Aspect-Based Sentiment Analysis. In particular, we intend to tackle the task of Aspect-Term Extraction, dealing with the problem of word feature, n-grams and custom feature combination to present a reliable and scalable model.

1.1.2 Specific Goals

1. Reviewing state-of-art works on Aspect-Based Sentiment Analysis focusing on supervised methods and feature engineering.

2. Identifying current problems on the existing methods in the state of art concerning

ABSA in terms of feature engineering and feature combinations, and perform a study on how to overcome these limitations in our proposal.

3. Implementing a computational solution based on a pipeline of several steps concerning text preprocessing, text annotation, feature engineering, and Aspect Term Extraction based on a supervised machine learning algorithm.

4. Performing several experimental evaluations on two benchmarking datasets from the SemEval 2014 ATE shared-tasks in order to answer some crucial experimental questions raised by this work.

5. Discussing and finding the best feature combination in terms of specific evaluation metrics used in Sentiment Analysis.

6. Carrying out a comparative assessment against the best supervised solutions to the same problem which were evaluated on the same reference (SemEval 2014) datasets.

1.2 Document Structure

Chapter 2 introduces the basic foundation necessary to the understanding of this document, this chapter also gives a review of concepts used and details tools and methods widely used on the literature that are part of the developed methods. Giving the definition for Natural Language processing, tools and applications. Also, associating Sentiment Analysis and Aspect-Based Sentiment Analysis to NLP, defining their main features, tasks, approaches and methods.

Chapter 3 reviews the Aspect-Based Sentiment Analysis and Aspect Category Detection literature. Discussing several works related to this project along with their main contribution to the field state of art. Analyzing works in the field of Aspect-Term Sentiment Analysis and the specific task of Aspect Term Extraction, listing and analyzing the main features, methods and datasets.

Chapter 4 describes the developed method, explaining the steps for dataset preprocessing, feature engineering, extraction and representation, and output file generation. Along with the details about the experiment configuration and training and testing execution. With details, figures and tables to detail the method chosen by this work.

Chapter 5 presents the results obtained by the developed method. Detailing the setup to the experiments with its datasets and annotation schema. Explaining the evaluation metrics and analyzing the results of the performance that the method applied in this work obtained. Also, the chapter presents a study on Feature Importance with the features used in this work, and the SemEval 2014 Official Results, since the datasets used were from

SemEval 2014. This chapter ends with the answers to the experimental questions that were made during this work.

Chapter 6 summarizes and discusses the contributions, shortcomings and open paths of exploration for further improvement of the present research work. Presenting the limitations that this work reached, the final analysis obtained from the results and new ideas to pursue in the future.

2 Theoretical Foundation

2.1 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science that deals with the interaction between human and computers through language. It is used to make human's natural language understandable to computers.

Early computational approaches to language research focused on automating the analysis of the linguistic structure of language and developing basic technologies such as machine translation, speech recognition, and speech synthesis. Today's researchers refine and make use of such tools in real-world applications, creating spoken dialogue systems and speech-to-speech translation engines, mining social media for information about health or finance, and identifying sentiment and emotion toward products and services.

2.1.1 NLP Tools

NLP Tools add non-explicit information to sentences, called annotations. This task demands computational work, varying from the complexity. Some frameworks for text processing are NLTK (LOPER; BIRD, 2002), GATE (CUNNINGHAM, 2002), and Stanford Core NLP Toolkit (MANNING et al., 2014), all of them aim to offer the main tools and methods to aid text processing.

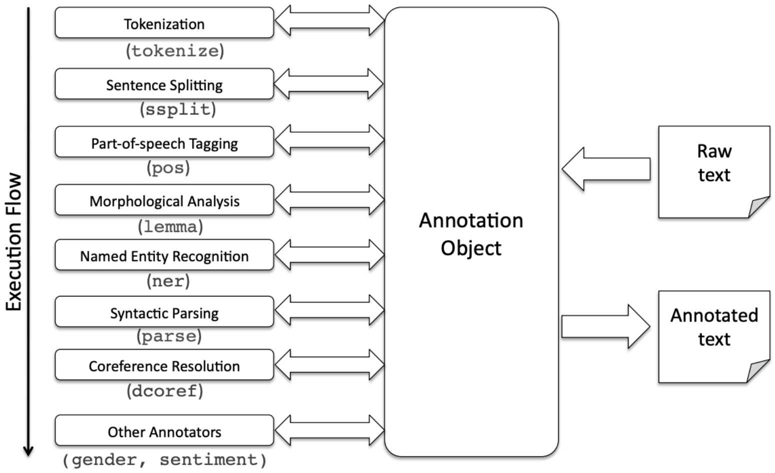


Figure 1 – Overall NLP Architecture

Source: Manning et al. (2014)

The system receives a review as input, represented by an Annotation Object that have been ran through many processing steps. These steps are called external resources due to not being explicit information on the text. An annotated review is generated as output. Figure 1 represents the main view of the Stanford NLP tools, one of the most used in literature. NLP works on many levels and for most of the time, those levels communicate with one another.

The **Morphological** level of linguistic processing is the level that studies word structure and word formation, with focus on the analysis of the individual components of words. The *morpheme* is the minimal unit of meaning, and is the most important unit for morphology. In this work, terms are stemmed to reach the morphological root of each terms so to be fed to the model, and will match other terms that might be in plural or other form.

The **Syntactic** level of linguistic processing can use the output of PoS tagging to group words into phrases. Also called *parsing*, Syntactic Analysis allows the extraction of phrases to give more meaning than it would be possible by using only the individual words. Parsing is usually leveraged to improve indexing, since phrases can be used as representation of documents, providing better information than just single-word indices.

The real meaning of the sentence is dealt by the **Semantic** level of linguistic processing, relating syntactic features and treating ambiguous words with multiple meanings to the given contexts. This level provides the correct interpretation of sentence meanings, in comparison to the analysis at word or phrase levels. The discourse level of linguistic processing addresses the analysis of structure and meaning of a text, making connections between words and sentences surpassing the analysis of a single sentence.

The **Lexical** analysis partitions the sentence in tokens, usually words. This is the most simple step, as most models use the space character as the separator, generating a sequence of tokens.

The **Grammar** analysis uses a process called PoS tagging, which determines the part of speech of each token (i.e. noun, verb, adverb). This method uses a combination of dictionaries and learning models, that have an elevated computational cost. Table 1 presents a regular set of PoS tags.

2.1.2 Applications

2.1.2.1 Machine Translation (MT)

Machine Translation software is a fully automated solution that aims to substitute words from one language to another, translating source content into target languages. Humans generally use MT to help translate text and speech into another language, or the MT software may operate without human intervention.

Name	Description	Name	Description
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non 3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh determiner
PDT	Predeterminer	WP	Wh pronoun
POS	Possessive ending	WP$	Possessive wh pronoun
PRP	Personal Pronoun	WRB	Wh adverb

Source: Santorini (1990)

Table 1 – PoS list of the model PENN TREEBANK PROJECT

MT tools are used to translate large amounts of information that are not feasible to be done manually. Sometimes the simple translation may not produce good results, hence the necessity of training in the desired domain and language. This problem is being tackled by works using corpus statistical and neural techniques.

Translation companies use MT to increase productivity of their translators or cut costs, going through a large growth in is use. Examples of MT are Google translate, Microsoft translator, and Bing.

2.1.2.2 Information Retrieval (IR)

Information Retrieval is an automatic process that receives an user query and responds by examining a collection of documents and returning a sorted document list relevant to the user requirements. Those queries are be based on metadata or full-text indexing. Information retrieval studies the representation, knowledge and search of relevant information from knowledge sources, and it is the main technology behind search engines.

2.1.2.3 Information Extraction (IE)

Information extraction (IE) is the automated method to retrieve information from a body of text. IE tools provide the possibility to extract information from documents, databases, sites or other sources, pulling this information from unstructured, semi-structured, structured or machine-readable text. The main use for IE in NLP is to extract structured text from unstructured documents.

IE relies on Named Entity Recognition (NER) to find targeted information on text, by using NET to recognize entities as categories.Once the category is retrieved, the named entity's information is extracted and turned into a machine-readable document, so algorithms can process to extract meaning. Other subtasks such as coreference resolution, relationship extraction, language, vocabulary analysis, and audio extraction are some of the ways IE can be use to find meaning.

2.1.2.4 Summarization

Text summarization is the technique of reducing text, with the intention of creating a fluent, short, and accurate summary using the main points outlined from a longer document. Summarization methods are needed to consume the increasing amount of text available online. Fundamentally, summarization's goal is to provide the ability to users to consume useful and relevant information in a fast manner.

Automatic summarization is used for many applications, depending on the use case and type of documents. Summarization systems can be categorized into Abstractive and Extrative.

Abstractive summarization can be mirrored to the human way of summarizing a corpus of text, by rewriting the main points using their own words. This technique requires high-level human skills such as the ability to combine different perspectives into coherent text. As 2019, abstractive summarization has not been satisfactory, so many systems for automatic summarization opt to use extractive summarization.

Extractive summarization is done by are excerpts taking excerpts directly from the input documents and presented in a readable manner, employing AI techniques to identify the most important sentences from the source. The summary does not contain any rewriting of the ideas from the original text.

2.2 Sentiment Analysis

Sentiment Analysis, also known as opinion mining, is a field of study which aims to extract opinions or sentiments from a text. This means that an analyst can extract information about the reach of a product or service only from user comments and reviews, without the need of forms or evaluation systems. It can additionally be used to retrieve opinions about politics, for after important events one can see many comments about the matter.

Opinions, as opposed to facts, are subjective expressions that express one's sentiment about a subject. Opinion mining is focused on analyzing opinions that express or imply positive and negative feelings, however in some cases, it's interesting to also process neutral information (LIU, 2012).

With the growing success of social media (Facebook, Instagram, Twitter), users around the world tend to comment on every matter happening every day, generating a considerable amount of data that needs to be analyzed, however it is unfeasible for a human being to accomplish. So that, Sentiment Analysis tools are increasingly necessary to obtain relevant information on users opinions in an efficient way.

In opinion mining, the indicators of sentiment are called opinion words. These words are associated with sentiments, for example: good, incredible, and amazing are associated with a positive sentiment; whereas bad, awful, and terrible can be tied to a negative opinion. These words are put together in a dictionary, called a opinion lexicon, that can be used to find words that carry sentiment. However, the opinion lexicon has its limitations, due to words having different, even opposite, meaning across domains.

In an opinion, the text is directed towards an entity, that can be a person, an object, its attributes or its features, for example, in the sentence: *The **pasta** here is **marvelous***; A positive opinion (marvelous) is expressed about the entity (pasta), this is an example of a direct opinion. Another type of opinion is comparative opinion, where two or more entities or attributes of entities are compare with one another. For example: *The ambiance in restaurant A is way better than in restaurant B.*

2.2.1 Sentiment Analysis Levels

The study of sentiment analysis is possible at three levels, as stated by Hu and Liu (2004), document level, sentence level and entity or aspect level.

The opinion classification in a document level generally is insufficient, due to its analysis being about the opinion in a high level to an entity, not taking into consideration individual aspects. Besides that, an opinion can, at the same time, evaluate several aspects of an entity. To consider a document level opinion as positive implicates that all aspects related to an entity are positive, which can be untrue (LIU, 2012).

Consider the following example on a sentence level: happy to meet you is considered a positive sentence, while My phone is very interesting but need enhancement in some issues is considered a positive in document level if the whole text was considered as one entity.

A difficulty that occurs with both document and sentence level analysis is confluence, where one unit of analysis (i.e., a document or sentence) contains polar opinions. To illustrate this phenomenon, the following sentence from a data set containing restaurant reviews, is shown below: “The food was good, but the service was bad.” In this review, the aspects ‘food’ and ‘service’ are mentioned with respectively a positive and a negative sentiment. As these polarities are conflicting, the sentiment of the sentence as a whole will be in the middle, in this case neutral. By labeling this sentence as neutral, it is suggested that there is no sentiment in this sentence, which is not the case (BAAS et al., 2019).

The aspect level can lead to a better analysis and results if taken into consideration. Consider the following example on the aspect level: My phone is really nice but I have a bad battery. It contains slow applications but I am happy with its screen. The aspect here is the phone while the attributes are battery, applications, and screen. Sentiment detection can lead to the following results (battery, negative), (application, negative), (screen, positive).

2.2.2 Aspect-Based Sentiment Analysis (ABSA)

Sentiment analysis is increasingly viewed as a vital task both from an academic and a commercial standpoint. The majority of current approaches, however, attempt to detect the overall polarity of a sentence, paragraph, or text span, regardless of the entities mentioned (e.g., laptops, restaurants) and their aspects (e.g., battery, screen; food, service). By contrast, ABSA’s goal is to identify the aspects of given target entities and the sentiment expressed towards each aspect.

Early work in sentiment analysis mainly aimed to detect overall polarity (e. g. positive, negative) of a given text span (PANG; LEE; VAITHYANATHAN, 2002); (TURNEY, 2002). However, the need for a more fine-grained approach, such as aspect-based (or feature-based) sentiment analysis (ABSA), soon became apparent (LIU, 2012).

Aspect-Based Sentiment Analysis’ main goal is to obtain the most detailed entities and aspects in a text. In this work, the dataset hierarchy is in the following order: the domain is a limited source of knowledge; a dataset is a set of reviews of a given domain; a review has one or more sentences; a sentence contains zero or more opinions.

2.2.2.1 Main Features

The phase of feature extraction is an important phase for the process of information extraction of a text. The features are the elements algorithms use as input data for training and classification.

The main features to be used in the development in this work are presented as follows:

1. **Tokenization**, as defined by Manning et al. (2014), is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters,

such as punctuation. These tokens are often loosely referred to as terms or words, but it is sometimes important to make a type/token distinction. A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A *type* is the class of all tokens containing the same character sequence.

2. **Stemming** is done by removing any attached suffixes and prefixes (affixes) from index terms before the actual assignment of the term to the index (JIVANI et al., 2011). Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction. Stemming and AI knowledge extract meaningful information from vast sources like big data or the Internet since additional forms of a word related to a subject may need to be searched to get the best results. Stemming is too a part of queries and Internet search engines.

3. **Lemmatization** is the process of finding an word that can represent every form of a word. The process is done with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*.

4. **PoS Tagging** is the process of correlation between a word in a sentence and its Part of Speech. It is based both on the word's definition and its relationship with other words that gives it context.

5. **Chunking** is the hierarchy of ideas in a text, it gives the ability to the speaker to generalize or specify a word inside a sentence. Chunking up or down allows the speaker to use certain language patterns, to utilize the natural internal process through language, to reach for higher meanings or search for more specific bits/portions of missing information.

Aspect-Based Sentiment Analysis can be divided in four subtasks, the are Aspect Term Extraction, Aspect Term Polarity, Aspect Category Detection, and Aspect Category Polarity.

2.2.2.2 Aspect Term Extraction

An entity of the target is defined by its particular aspect term. The term is defined by its unique position in a text. It may not be explicit, and can be expressed by pronouns or text coreferences. For explicit target extraction, there are four main approaches:

1. **Noun-based extraction:** Initially developed by Hu and Liu (2004), uses a grammar analyzer to identify the most frequent nouns.

2. **Sentiment and target relation extraction:** Uses a grammar analyzer and dependency relations to find relation between sentiment words and their targets.
3. **Supervised learning extraction:** Uses supervised machine learning models to determine if an opinion is about an entity or an aspect of an entity.
4. **Topic model extraction:** Uses cluster-based methods in order to obtain distributions that represent aspects.

2.2.2.3 Aspect Term Polarity

Each opinion in a sentence has a polarity originated from the set P = *{positive, negative, neutral}*. A neutral sentiment classification happens when there is no clear definition about its polarity. Two main approaches are used to determine an opinion's polarity:

1. **Supervised learning based attribution:** Is an approach that uses supervised learning in sentence level to determine a sentence's opinion. The sentence can be the scope of the sentiment expression. This approach makes the method dependent on the training data, yielding poorer results when applied to different domains (LIU, 2012).
2. **Lexical information based attribution:** Is a set of methods usually supervised that use opinion dictionaries and processing resources such as grammar analyzers or dependency trees to determine an opinion's polarity.

2.2.2.4 Aspect Category Detection (ACD)

Aspect Category Detection, or Aspect Category Classification, is concerned about identification of associated entities and attributes, both implicit and explicit (DOHAIHA et al., 2018). In the sentence *It has great **sushi** and even better **service**.*, given the predefined categories, the task is to identify the entity - the aspect of *sushi* as **food** and an attribute denoting **quality**; and the aspect of *service* as **service** and the attribute **general**.

The task for Aspect Category Detection is to identify aspect categories discussed in a predefined set of aspect categories *(e.g., PRICE, FOOD)* within a set of sentences. Aspect categories are typically coarser than the aspect terms and they do not necessarily occur as terms in the sentences (PONTIKI et al., 2014). For example, in *Delicious but expensive*, the aspect categories **FOOD** and **PRICE** are not instantiated through specific aspect terms, but are only inferred through the adjectives **delicious** and **expensive**.

Each category is unique in a said domain, and this makes the use of dictionary insufficient to identify synonyms in different domains. The necessity of prior knowledge about a said domain adds difficulty to the task of Category Detection. The main strategies use supervised methods, non-supervised methods and association rules (LIU, 2012).

2.2.2.5 Aspect Category Polarity

Given a predefined set of aspect categories, Aspect Category Polarity aims to classify the polarity of each aspect in a sentence. Taking the example sentence: *The pasta was amazing, but the lights were not very good.*, where the word *pasta* is categorized as *food* with the polarity **positive**, and the word *lights* is categorized as *ambiance* and has a **negative** polarity.

2.2.3 Main Approaches

For Sentiment Analysis, the two main approaches are Machine Learning and Lexicon-based approaches, as shown by Figure 2.

Lexicon-based approaches use a list of aspect-related sentiment phrases as the core resource (DING; LIU; YU, 2008); (HU; LIU, 2004), and are made from the polarity calculation of a review using the semantic orientation of words or sentences. The dictionary-based method uses a collection of opinion words, together with a positive or negative marking to determine the sentiment.

Machine learning methods use learning algorithms to determine the sentiment without depending on a database of words, making it faster than other methods. The key issue for learning methods is to determine the scope of each sentiment expression, i.e., whether it covers the aspect in the sentence. (JIANG et al., 2011); (BOIY; MOENS, 2009)

2.2.3.1 Lexicon Based

The main principle to the Lexicon Based Approach, also known as Dictionary Based, is the use of lexicons, which are compilations of words or expressions of sentiment associated to its respective polarity (BECKER; TUMITAN, 2013).

One of the most used methods for the lexicon based approach is the joint occurrence between target and sentiment, in which a bag of words is used, not taking into consideration neither the order of the terms nor its syntactic relations. This method is mostly used to correlate a sentiment to an entity within a sentence. By taking the sentence *The pasta is awful*, the negative polarity of the word **awful** is associated to the entity **pasta**. The co-occurrence method yields good results when it is applied to shorter sentences (i.e. tweets, comments), because the sentiment words is close to the entity. When applied to a coarser level, it is necessary to establish an average over the sentiment words found.

Equation 2.1 sets an example to polarity determination of a document *"D"*, being *"Sw"* the polarity of a word *"w"* in a lexicon. The *weight()* function can be some distance

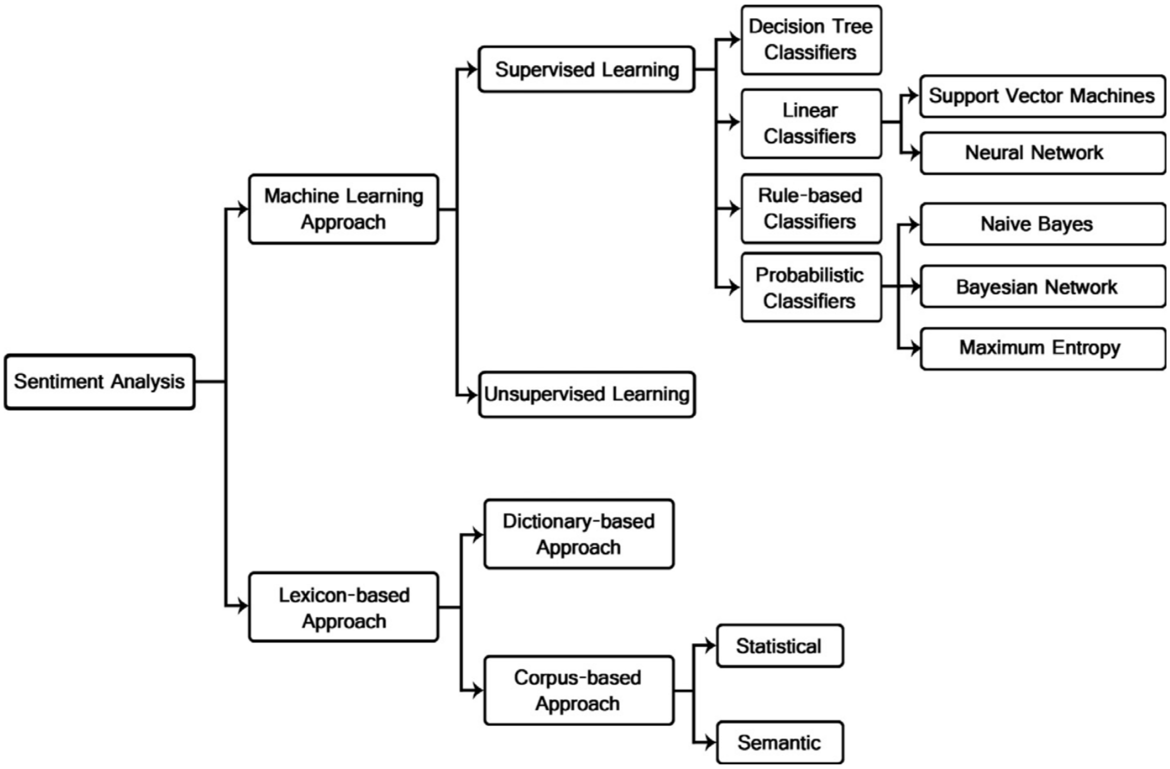


Figure 2 – Main approaches for sentiment analysis

Source: Medhat, Hassan and Korashy (2014)

measure between the sentiment word and the target. The *modifier()* function cant be used to treat negation words.

$$S(D) = \frac{\sum_{w \in_D} Sw \cdot weight(w) \cdot modifier(w)}{\sum weight(w)} \tag{2.1}$$

2.2.3.2 Supervised Machine Learning

Machine learning based approaches aim mainly in automatically discover general rules in large datasets, that allow information extraction that are implicit, and apply the new found knowledge into prediction of new information in a new dataset.

Machine Learning models have two common categorizations, Generative and Discriminative. Discriminative classifiers model the decision boundary between the different classes. Generative models, on the other hand, model how the data was generated, which after having learned, can be used to make classifications. As a simple example, Naïve Bayes, a very simple and popular probabilistic classifier, is a Generative algorithm, and Logistic Regression, which is a classifier based on Maximum Likelihood estimation, is a discriminative model.

Another categorization for machine learning models are the two main approaches:

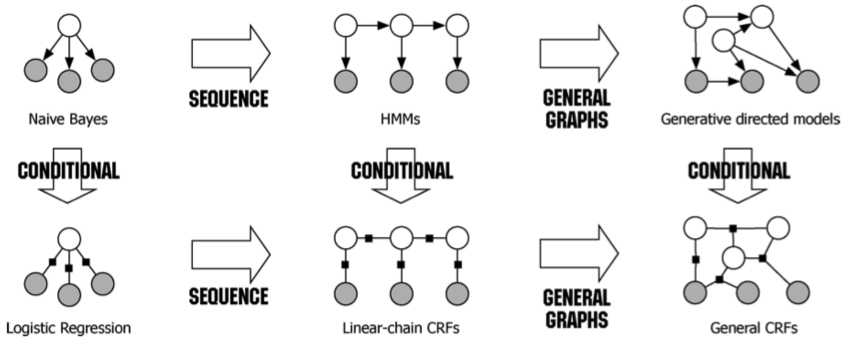


Figure 3 – Diagram of the relationship between Naïve Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

Source: Sutton, McCallum et al. (2012)

Unsupervised Learning Approaches which use learning patterns in the input when there are not specific output values. Unsupervised Methods can be used to label a corpus that can be used afterwards by a supervised learning classifier. K-means is a well known example of unsupervised learning algorithms. **Supervised Learning Approach**, on the other hand, depend on labelled corpora to train, and learn functions from examples of inputs and outputs. The output is a continuous value (Regression) or can predict a category or label of the input object (Classification). A good example of supervised learning is Naïve Bayes.

2.2.3.3 Conditional Random Fields (CRF)

Conditional Random Fields (CRF), Figure 3, are a standard, discriminative model, and are used to predict the most likely sequence of labels that correspond to a sequence of inputs. Conditional Random Fields are a type of Discriminative classifier, and as such, they model the decision boundary between the different classes (CHAWLA, 2017).

CRF's inherent principle is that Logistic Regression is applied on sequential inputs. These models are a way to combine advantages of discriminative classification and graphic modelling, combining the ability to compactly model multivariate outputs $(\mathbf{y})$ with the capacity of leveraging a large number of prevision input resources $(\mathbf{x})$.

An advantage of a conditional model is that of its dependencies that have only $(\mathbf{x})$ variables, that do not play any role in the conditional model, meaning that a precise conditional model can have a much simpler structure than an ordinary one. CRF presents another advantage, since it computes the joint probability distribution of the entire label sequence when an observation sequence intended for labeling is available, rather than defining the state distribution of the next state under the current state conditions given. The difference between generative models and CRFs is similar to the difference between Naïve Bayes and logistic regression classifiers. The multinomial logistic regression model can be seen as a simpler type of CRF, in which there is only one output variable.

One problem for CRF is the one of exact inference, for general graphs, as it is unmanageable for CRF. In that case, this problem can be attacked by using algorithm that apply message passing algorithms, if the graph is a chain or a tree. Other possible solution is the use of combinatorial min cut/max flow algorithms can yield exact results when the CRF only contains pair-wise potentials and the energy is sub-modular. If it is impossible to accomplish inference, some algorithm such as Loopy belief propagation, Alpha expansion, Mean field inference and Linear programming relaxations can be applied to achieve approximate solutions. Other problem that can be seen is that CRF is highly computationally complex at the training stage of the algorithm. It makes it very difficult to re-train the model when newer data becomes available.

CRF is described by Equation 2.2, where Y is the hidden state and X is the observed variable. There are two components to the CRF formula:

1. **Normalization**: There are no probabilities on the weights and features (right side). However, the output is expected to be a probability and hence there is a need for normalization. The normalization constant Z(x) is a sum of all possible state sequences such that the total becomes 1.

2. **Weights and Features**: This component can be thought of as the logistic regression formula with weights and the corresponding features. The weight estimation is performed by maximum likelihood estimation and the features are previously defined

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right\} \tag{2.2}$$

3 Related Work

This chapter's aim is to briefly review the literature in Aspect-Based Sentiment Analysis and Aspect Category Detection. Several works, in which this project is based upon, are discussed along with their main contributions to the development of the field.

The following sections are organized as follows: Section 3.1 details studies focused on Aspect-Based Sentiment Analysis, Section 3.2 approaches works in the field of Aspect Term Extraction, and Section 3.3 summarizes the state of art that this work is based upon.

3.1 Aspect-Based Sentiment Analysis

Zhou, Wan and Xiao (2015) propose a representation learning approach to automatically learn useful features for Aspect Category Detection. Specifically, a semi-supervised word embedding algorithm is first proposed to obtain continuous word representations on a large set of reviews with noisy labels. Afterwards, the authors generate deeper and hybrid features through neural networks stacked on the word vectors. A logistic regression classifier is finally trained with the hybrid features to predict the aspect category. The experiments are carried out on the restaurant review dataset released by SemEval-2014 and achieved a F-1 score of 90.10%, outperforming the main systems of the event.

The system proposed by Machacek (2016) models the task of Aspect Category Detecion as a multi-label classification with binary relevance transformation, where labels correspond to the entity-aspect pairs. The author published his system running in two models: Constrained (using no external data sources such as lexicons or additional training sets) and Unconstrained (no data source restriction). Words from each sentence are used as individual binary features of that sentence. For each entity-aspect pair, all training sentences are used as positive or negative examples of that entity-aspect pair. Vowpal Wabbit1, a supervised machine learning tool, is used to train the resulting binary classifiers. More precisely, a variant of online gradient descent algorithm is used to perform logistic regression with squared cost function. The systems uses features such as Lemma, PoS, Token, Stop words, Prices recognition, N-grams, Minimal word length, and Consecutive letters neutralization. The system proposed by this study achieved a F1-score of 71.49% for restaurant domain and 47.52% for laptop reviews using SemEval 2016 datasets.

Kauer (2016) proposes two methods directed towards two fundamental points to opinion treatment: aspect-based sentiment analysis, which identifies expressions mention aspects and entities in a text, using natural language processing tools combined with machine learning algorithm; and polarity attribution, which uses twenty-four features extracted from a search engine generated ranking in order to produce machine learning models. Besides that, the methods do not depend on linguistic resources and can be applied over noisy data. The experiments were done in order to fulfill SemEval 2015 Aspect-Based Sentiment Analysis tasks. The author proposed the classifier Simple Logistic (LANDWEHR; HALL; FRANK, 2005) which utilizes logistic linear regression models, using word features such as token, sentence split, PoS, Lemma, Dependency Parsing, and Coreference. The system achieved a F-1 score of 44.95% for Laptops domain and 51.88% for Restaurants domain.

Kok et al. (2018) considers the task of aspect-based sentiment analysis at the review-level for restaurant reviews. The authors focus on ontology-enhanced methods that complement a standard machine learning algorithm. For this task, two different algorithms are used: a review-based and a sentence aggregation algorithm. By using an ontology as a knowledge base, the classification performance of the models improves significantly. Furthermore, the review-based algorithm gives more accurate predictions than the sentence aggregation algorithm. The dataset used was SemEval 2016 ABSA task restaurant reviews. For the sentiment classification the authors use a linear Support Vector Machine (SVM). For the review level a multiclass SVM model is trained with the classes: positive, negative, neutral, and conflict. For the sentence level a multiclass SVM model is trained with the classes: positive, negative, and neutral. The SVM models uses a variety of features to determine the sentiment classification, these features can be split into two groups: generated features, such as Aspect, sentence count, lemma, ontology concepts, and sentiment count; and adapted features, for instance ontology concept score, negation handling, synonyms, weights, and word window. The system achieved a result of 87.18% for restaurant domain.

Baas et al. (2019) proposed a method using a Support Vector Machine with the libSVM library setting 6 different pattern classes: lexical, syntactical, semantic, sentiment, hybrid, and surface for the Sentiment Polarity Classification task in Aspect-Based Sentiment Analysis, using the SemEval 2015 restaurant and laptop domains datasets. Showing that several of these patterns, including synset bigram, negator-POS bigram, and POS bigram, can be used to better determine the aspect-based sentiment, using two widely used real-world data sets on consumer reviews. Features such as Word n-grams (unigram to fourgram); PoS n-grams (unigram to fourgram); Synset n-grams (unigram and bigram); Synset-PoS bigram; Negator-PoS bigram; Sentisynset unigram; Negator-sentisynset bigram are used. The proposed approach achieves 69.0% and 73.1% F1 score for the two data sets, respectively.

Movahedi et al. (2019) model utilizes several attentions with different topic contexts, enabling it to attend to different parts of a review sentence based on different topics. The authors proposed a new neural architecture, Topic-Attention Network (TAN) to capture important words given different topics and applied it to the restaurant reviews in SemEval 2014 and 2016 datasets. Second, by converting the problem into a vector space using the squash activation function (Sabour, Frosst, and Hinton 2017) and treating the length of the output vectors as probabilities, they show the effectiveness of the squash function in the aspect category detection. The system obtained a F1-score of 90.61% for 2014 datasets and 78.38% for 2016 datasets.

The study by Xia et al. (2019) proposes an approach for online review sentiment classification using a Conditional Random Field algorithm to extract the emotional characteristics from fragments of the review in counterpart to sentiment analysis based on a domain dictionary, that relies on the comprehensiveness and accuracy of the dictionary. The characteristic (feature) words are then weighted asymmetrically before a support vector machine classifier is used to obtain the sentiment orientation of the review. The authors used 1488 preprocessed online Chinese reviews Audi A4 sedan from <www.autohome.com.cn> divided in two sets A and B, and 1061 English online reviews of a screen protector for the Samsung Galaxy S7 from <www.amazon.co.uk>, likewise divided in two sets B and C. The study presented three experiment: Experiment 1 uses the manually annotated CRF features for the A and C review sets, after which a CRF model trained using the A and C review sets was used to annotate the B and D review sets and extract the sentiment feature fragments. Finally, employing asymmetric weights to classify the emotional feature fragments of the B and D review sets using an SVM. In experiment 2, the B and D review sets were used as data sources. In experiment 3, the B and D review sets obtained from Experiment 1 were used as data sources.

The proposed system first resulted in a accuracy of 68% for the Chinese reviews. Then, by using the CRF model to extract the emotional feature fragments increased the average accuracy 78%. The same process was applied to the English reviews, which resulted in an average accuracy of 80% without the CRF model and 91% when the CRF model was used. Comparing the results of Experiments 1 and 3, the CRF model was used to extract the sentiment feature fragments, TF–IDF was used to assign weights to the words, and an SVM was used to classify the Chinese reviews, resulting in an average accuracy of only 78%. However, the use of asymmetric weights increased the average accuracy to 90%. The same process was applied to the English reviews, which resulted in an average accuracy of 91%, irrespective of whether the asymmetric weights were used.

3.2 Aspect Term Extraction

Aspect extraction may be considered as a sequence labeling task because the product aspects occur at a sequence in a sentence (ZHANG; LIU, 2014).

The study done by Chernyshevich (2014) focused in cross-domain extraction of product features using CRF, and aimed to fulfill the phrase-level sentiment classification, namely aspect extraction. Her system is based on IHS Goldfire linguistic processor and uses a rich set of lexical, syntactic and statistical features in a CRF model. The system is trained on mixed training data, and the same model was used unchanged for classification of both domain-specific test datasets. Instead of using the Inside-outside-begins (IOB) notation, the author decided to introduce new labels: **FA** for the attribute word preceding head word of a noun group; **FH** for the head word of a noun group; **FPA** for attribute word after head word of a noun group, and **O** for other non-aspect tokens. The study's experiments showed that the words used in aspect terms are easier to recognize when they are always tagged with the same tags. The features were used in the CRF model to the current token, two previous and two next tokens, some of them are: token, PoS, NER, semantic category, semantic orientation, frequency of token occurrence, opinion target, noun phrase features and Subject-Action-Object (SAO) features. This system achieved a F-1 score of 0.7962 for the restaurant domain, and 0.7455 for the laptop domain.

Toh and Wang (2014) proposed a system that consists of two components to address two of the SemEval 2014 Aspect-based Sentiment Analysis subtasks respectively: a Conditional Random Field (CRF) based classifier for Aspect Term Extraction (ATE) and a linear classifier for Aspect Term Polarity Classification (ATP). For the ATE subtask, the authors implement a variety of lexicon, syntactic and semantic features, as well as cluster features induced from unlabeled data. They implement a set of general features, such as token, PoS, Head word, Dependency Relation, and Name list. Additional features that require external resources and/or complex processing are also used, such as WordNet Taxonomy, Word Cluster, and Name List Generated using Double Propagation. For each domain, the authors make submissions in both constrained and unconstrained settings. Using the optimum feature set found by applying 5-fold cross-validation, separate models were trained for each domain (Restaurants and Laptops) and were evaluated against the SemEval-2014 ABSA Task, obtaining results of 70.41% for Restaurants constrained and 73.78% for unconstrained; for laptops, the results were 78.34% for constrained and 84.01% for unconstrained. in Restaurants and Laptops domains.

The opinion mining system Sentiue, proposed by Saias (2015) applies a supervised machine learning classifier, for each label, followed by a selection based on the probability of the entity/attribute pair for Aspect Category Detection. For Aspect Term Extraction, the system uses a catalog of known targets for each entity type, complemented with named entity recognition. In Sentiment Polarity Classification, the author used a 3 class polarity classifier, having BoW, lemmas, bigrams after verbs, presence of polarized terms, and punctuation based features. Working in unconstrained mode, the system's results for ACD were assessed with precision between 57% and 63%, and recall varying between 42% and 47%. In SPC, Sentiue's result accuracy was approximately 79%, reaching the best score in 2 of the 3 domains.

The work done by Kumar et al. (2016) achieved best results in sentiment polarity classification about English laptops, Spanish restaurants and Turkish restaurants, and Scored second for English restaurants. It aimed to fulfill SemEval 2016 task 5 subtastks: Aspect Category Detection, Opinion Target Extraction, Sentiment Polarity Classification, covering the languages English, Spanish, Dutch, French, Turkish, and Arabic; and proposed two classifiers: SVM for Aspect Category Detection and Sentiment Polarity Classification, and CRF for Opinion Target Extraction. They used a set of word features such as Lemma, PoS, Chunk, Named entity information, and a set of syntactic features such as WordNet, Prefix and suffix, tf-idf, and Bag of words. The system got a significant improvement on adding information from the induced lexicons in each language.

Ruder, Ghaffari and Breslin (2016) proposes a Convolutional Neural Network (CNN) for both aspect extraction and aspect-based sentiment analysis, working on multiple language ABSA tasks, fulfilling the subtasks of Aspect Term Extraction and Aspect Term Polarity Classification on the domains Restaurants, Laptops, Phones, and Hotels. The proposal casts aspect extraction as a multi-label classification problem, outputting probabilities over aspects parameterized by a threshold. To determine the sentiment towards an aspect, the authors concatenate an aspect vector with every word embedding and apply a convolution over it. The proposed work achieved convincing results in the multilingual setting, which is particularly appropriate for neural networks due to their language and domain independence, the best results were for restaurant domain of 68.10% of F-1 score.

The system proposed by Toh and Su (2016) was submitted to the subtasks of Aspect Category Detection and Opinion Target Extraction of SemEval 2016. It consists of two components: binary classifiers trained using single layer feedforward network for aspect category classification, and sequential labeling classifiers for opinion target extraction. Besides extracting a variety of lexicon features, syntactic features, and cluster features, the authors explore the use of deep learning systems to provide additional neural network features. Some of the features used were Token, Name list, Head Word, Word embeddings, and Word Cluster. In the task of Aspect Category Detection, for each category found in the training data, a binary classifier is trained using the Vowpal Wabbit tool, which provides the implementation of the single layer feedforward network algorithm that is used. Opinion Target Extraction is treated as a sequential labeling task. The sequential labeling classifiers are trained using Conditional Random Fields (CRF). The implementation of CRF is provided by the CRFsuite tool. The system participated in both unconstrained and constrained settings for the English datasets, it ranks first for all four evaluations that it participated, achieving F-1 score of 73.03% for restaurants and 51.94% for laptops.

Study	Main Features	Classifiers	Domain	Tasks	Results
Chernyshevich (2014)	Token; PoS; NER; Semantic Category; Semantic Orientation; Frequency of token occurrence; Noun phrase features; SAO	CRF	Restaurants; Laptops	ATE	R: 0.7962; L: 0.7455
Toh and Wang (2014)	Word; PoS; Head Word; Head Word PoS; Dependency Relation; Name List; WordNet Taxonomy; Word Cluster; Name List;	CRF; Linear Classifier	Restaurants; Laptops	ATE; SPC	R: 0.7041; L: 0.7834
Zhou, Wan and Xiao (2015)	General features and aspect-specific features learned through two different kinds of Neural Network	Logistic Regression Classifier	Restaurants	ACD	R: 0.9010
Saias (2015)	BoW; Lemmas; Bigram; Presence of negation terms; Bigram after negation term; Presence of exclamation/question mark; Presence of polarized terms;	Maximum Entropy	Restaurants; Laptops; Hotels	ACD; ATE; SPC;	R: 0,834; L: 0,860; H: 0,863
Kumar et al. (2016)	Lemma; PoS; PoS+2; PoS-2; Chunk; Named entity information; Head Word and its PoS; WordNet; Prefix and suffix; tf-idf; Bag of words	SVM (ACD); CRF (OTE)	Restaurants; Laptops; Phones; Hotels	ACD; OTE; SPC	en: 0.6845; nl: 0.6437; es: 0.6973; fr: 0.6964
Ruder, Ghaffari and Breslin (2016)	Token	Deep Learning	Restaurants; Laptops; Phones; Hotels	ATE; SPC	R: 0.6810
Machacek (2016)	Lemma; PoS; Token; Stop words; Prices recognition; N-grams; Minimal word length; Consecutive letters neutralization;	Vowpal Wabbit1	Restaurants; Laptops	ACD	R: 0.7149 L: 0.4752
Toh and Su (2016)	Word; Name list; Head Word; Word Cluster	CRF; RNN	Restaurants; Laptops	ACD; ATE	R: 0.7303; L: 0.5194
Kauer (2016)	Token; Sentence Split; PoS; Lemma; Dependency Parsing; Sentiment Analysis; Coreference	Simple Logistic	Restaurants; Laptops	ATE; SPC	R: 0.5188; L: 0.4495
Kok et al. (2018)	Aspect; Sentence count; Lemma; Ontology Concepts; Sentiment Count;	SVM	Restaurants	ACD; SPC	R: 0.8718
Baas et al. (2019)	N-grams (1 to 4); PoS n-grams (1 to 4); Synset n-grams(1 to 2); Synset-PoS bigram; Negator-PoS bigram; Sentisynset unigram; Negator-sentisynset bigram	SVM with libSVM	Restaurants; Laptops	SPC	R: 0.6900; L: 0.7310
Movahedi et al. (2019)	Automatic generated features	Neural Network	Restaurants	ACD	2014: 0.9061; 2016: 0.7838
Xia et al. (2019)	Unigrams; Generated Emotional Feature Fragments	CRF; SVM	Cars and screen protector reviews	SPC	Acc: 0.9000 (CH); 0.9100 (EN)

Source: The author

Table 2 – Summary of the literature

3.3 Summary

Table 2 summarizes the state of art used in this work, listing the works and their main features, classifiers, domains, tasks and results. One can observe that the majority of works are based on the supervised approach, applying word features such as token and PoS. In more recent works, ATE systems have increased performance using more complex neural networks.

Concerning the features used, several works employed word features such as Token, PoS and Lemma with intent to extract aspects. For the models, CRF and SVM were the most used by the studies analyzed, but Deep Learning approaches were used additionally, producing satisfactory results.

The column “Main Features” shows that, in most of the work analyzed, information about tokens, PoS tagging and lexical-based features were the most employed in ABSA. However it is valuable to analyze features that are not so common, but brought satisfactory results when used, those are the n-grams and its combinations, as it is described in the work proposed by Baas et al. (2019).

The following column, “Classifiers” shows the classifiers selected by the works analysed. CRF was the most used, especially with the Aspect Term Extraction task, followed by SVM. Neural Networks and Deep Learning also were used by some works and achieved satisfactory results. Lastly, some custom classifiers were also used.

The column "Domain" describes which domains were used, where Restaurant and Laptops reviews being the most common. Hotels and Phones reviews also appeared in some works, while other works employed reviews for Cars and screen protectors.

The column "Tasks", shows the tasks each work accomplished, Aspect Term Extraction was the most applied to, followed by Aspect Category Detection and Sentiment Polarity Categorization. The last column, "Results", presents the results obtained by each work, showing F-1 score or Accuracy (Acc) separating by the domains, languages or year of dataset.

After reviewing the works related to this project, one can observe that the works presented by Toh and Wang (2014) and Kauer (2016) are the ones with most similar approaches to this work, since they also adopted the CRF supervised classification algorithm. Concerning the features, our work was inspired from the recent work of Baas et al. (2019) which employed a feature engineering step mainly based on bigrams and synsets, but they performed a different task.

The approaches studied that were more refined suggest the use of dependencies to the application of the ABSA methods and its sub-tasks, mostly focused on the inter-relation between text components, and a specific aspect analysis, identifying and classifying pieces of sentences into categories, and their dependency relations. These tasks were performed with the use of different classifiers ranging from supervised to unsupervised machine learning algorithms, supervised methods being the most used and CRF the most common between the supervised classifiers. This gives these approaches a deeper touch, consisting of qualitative characteristics.

During the research for the development of this work, from the study on the literature, it was perceived that this work could contribute by performing a more detailed study on features in order to try to find the optimal combination of features for Aspect Term Extraction using CRF, by experimenting with feature combinations including word features such as tokens and PoS, n-gram features and resource-based or custom-made features. Such features included bigrams and resource-based features. Finally, this work also adopts CRF++ as its supervised classification algorithm due to its capability of providing an easier way to define, interpret, and combine several features.

4 Proposed Method

This section describes the developed method, explaining the steps for dataset preprocessing, feature engineering, extraction and representation, and output file generation. Along with the details about the experiment configuration and training and testing execution.

The method described in this section is applied to the datasets provided by SemEval 2014 Aspect-Based Sentiment analysis task, and it is developed to perform the Aspect Term Extraction task. In the edition of 2014, tasks were presented to extract aspect categories, sentiment polarity and aspect term extraction. The method presented in this work is developed by applying a series of steps to the dataset until reaching the final objective, the results. This work presents a solution using the approach of supervised learning, due to its knowledge base and baseline results in the field. The CRF classification method is used due to the wide documentation and ease of use in the task this work is set to perform

4.1 Functional Architecture of the Method

The method consists of 5 steps, as it is shown in Figure 4. The first step is the retrieval of data. The datasets used are SemEval 2014 task 4 restaurants and laptops reviews. Those reviews are annotated and indicate which aspect terms each review refers to.

The second step is preprocessing. Subsection 4.1.1 details how it is done. The task is to generate IOB (In-out-begin) labels for each word in each sentence in order to feed to the classifier. Together with the generation of labels, the third step is feature engineering, described in Subsection 4.1.2, which aims to extract feature combinations that can help achieve the best performance in this work's task.

After the first three steps are done, a set of features is generated together with the labels for the data and saved in an output file that is described in Subsubsection 4.1.2.2 to be fed to the CRF training algorithm. After that, the CRF trains and generates the model to make the predictions, Subsection 4.2.2 describes the training step. The test is done using a file with the same structure of the training file except for the last column, the label column, and the model generated by the training, Subsection 4.2.2 explains this step.

4.1.1 Dataset Preprocessing

The training file in xml format contains 3041 sentences for the restaurants domain and 3045 for the laptop domain, all previously annotated with a specific tag. The first

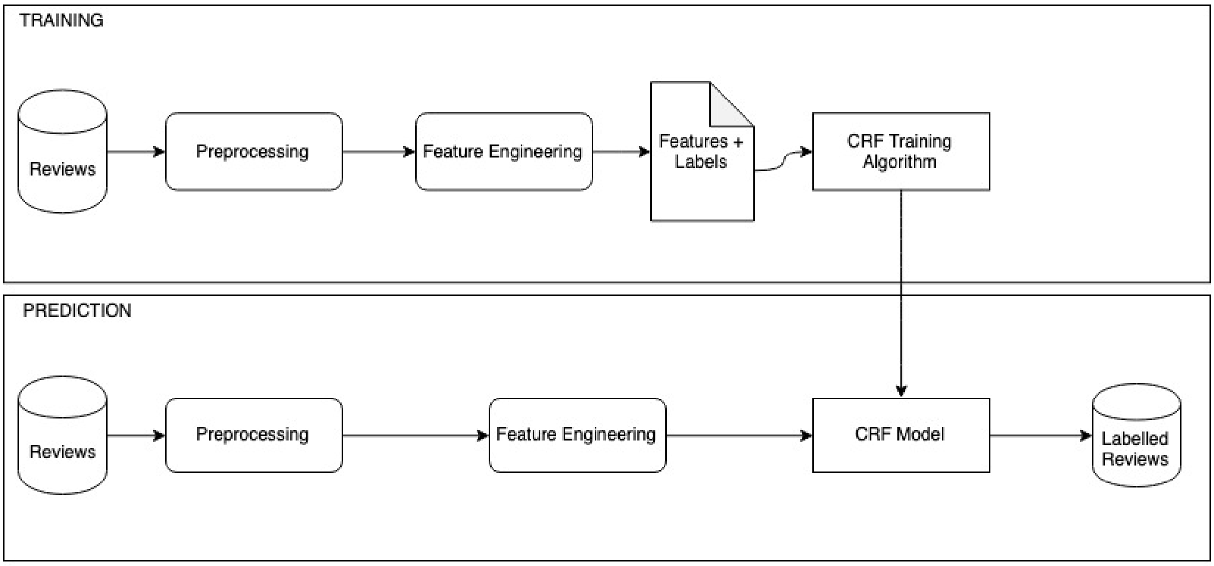


Figure 4 – Architecture of the method

Source: The Author

task was to bring the sentences to a Python environment, for that the library *ElementTree* [1] was used.

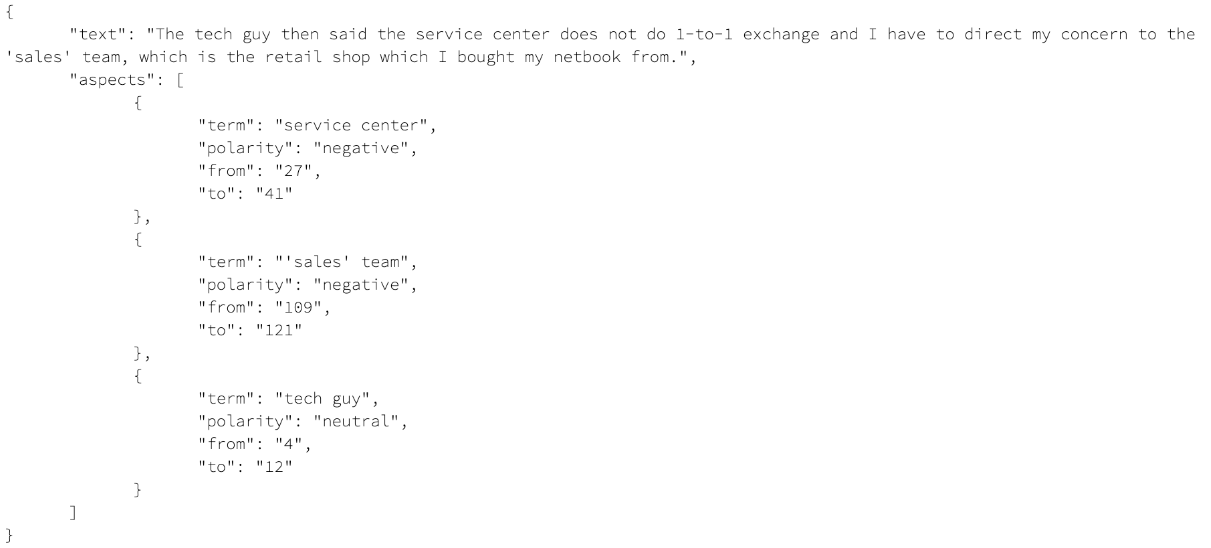
ElementTree is a simple object container designed to store hierarchical data structures on the memory (Python Software Foundation, 2019). The element type can be described as a Python's list of dictionaries. Thus, a vector contained all the sentences from the XML sentences is generated, where each sentence can be accessed and individually manipulated.

The data preprocessing is done in three steps described as follows:

The first step is use ElementTree to extract the root of the XML tree, then an empty list is declared where the dictionaries that represent the sentences objects will be saved. The root of the XML tree is a list, so each element is looped trough, saving the text in the object. The text is represented by the *<text></text>* tag, as it can be seen in Figure 10. After saving the text, it is necessary to loop in each element of the aspect terms object, which is also a list, this object is represented by *<aspectTerms></aspectTerms>*, then extract the attributes where the aspect term is saved. Finally, a dictionary list containing each sentence and its aspects is returned. Figure 5 displays an example of the extracted sentence object.

The second step is to extract the features for each sentence and save them in the output file, this will be explained with details in Subsection 4.1.2. The third step is to generate the labels for each word using the **IOB** (In-out-begin) notation. For this task, each word goes through a check, being compared to each aspect term that came from the

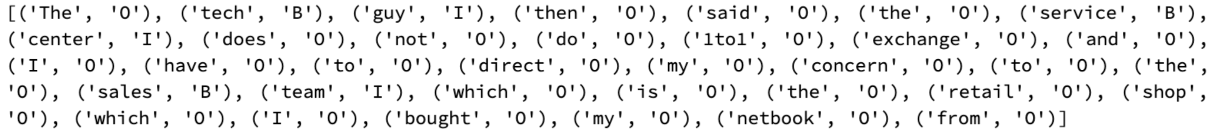
[1] https://docs.python.org/2/library/xml.etree.elementtree.html



```
{
	"text": "The tech guy then said the service center does not do 1-to-1 exchange and I have to direct my concern to the
'sales' team, which is the retail shop which I bought my netbook from.",
	"aspects": [
		{
			"term": "service center",
			"polarity": "negative",
			"from": "27",
			"to": "41"
		},
		{
			"term": "'sales' team",
			"polarity": "negative",
			"from": "109",
			"to": "121"
		},
		{
			"term": "tech guy",
			"polarity": "neutral",
			"from": "4",
			"to": "12"
		}
	]
}
```

Figure 5 – A sentence and its aspects extracted from the XML dataset and saved in a dictionary list to be easily manipulated

Source: The Author



```
[('The', 'O'), ('tech', 'B'), ('guy', 'I'), ('then', 'O'), ('said', 'O'), ('the', 'O'), ('service', 'B'),
('center', 'I'), ('does', 'O'), ('not', 'O'), ('do', 'O'), ('1to1', 'O'), ('exchange', 'O'), ('and', 'O'),
('I', 'O'), ('have', 'O'), ('to', 'O'), ('direct', 'O'), ('my', 'O'), ('concern', 'O'), ('to', 'O'), ('the',
'O'), ('sales', 'B'), ('team', 'I'), ('which', 'O'), ('is', 'O'), ('the', 'O'), ('retail', 'O'), ('shop',
'O'), ('which', 'O'), ('I', 'O'), ('bought', 'O'), ('my', 'O'), ('netbook', 'O'), ('from', 'O')]
```

Figure 6 – Labelled sentence using IOB notation

Source: The Author

annotated data. Sentences without aspect terms have every word is labelled as **O** (out). For sentences that have aspect terms, each word is compared to each aspect term so it can be labelled as **B** (begin) or **I** (in). Aspect terms may be composed of two or more words, because of that, they are split into lists and each item is compared to each word of the sentence, if the word matches the first item, it is labelled as B, if the word matches any of the other items, it is labelled as I. At end, a tuple containing each word and its respective label is generated for each sentence, Figure 6 shows an example of labelled sentence after the third step of preprocessing.

4.1.2 Feature Engineering

The phase of feature engineering is an important activity for Sentiment Analysis, in this phase all useful information is extracted from the text and its sentences, then fed to the classifier for training and posterior classification. Therefore, a part of this work is dedicated to the study and extraction of the main features used in Aspect-Based Sentiment Analysis, main features used in CRF and other features studied that can contribute to the solution proposed. The next subsections explain how the features are extracted and selected, and how the output file is generated to be used in the classifier.

4.1.2.1 Feature Extraction and Representation

The decision to use the following NLP tools in this work is due to the fact that they are not only the most used in the literature but also has state-of-the-art performance in the NLP task used by this work, also due to their robustness in quantity of corpora, libraries, ease of use and vast documentation combined with the previous familiarity with the Python language in which the tools are implemented.

1. **Tokenization** is taking a text or set of text and breaking it up into its individual words and punctuation. For this phase, NLTK was used. NLTK is a leading platform for building Python programs to work with human language data (NLTK Project, 2019). It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and wrappers for industrial-strength NLP libraries (LOPER; BIRD, 2002). Upon the application of tokenization in each sentence vector, tuples were generated containing each word of the sentences. Punctuation marks such as ".", ";", "?", "!" were removed. In addition to the current word, word bigrams are used as features.
2. **PoS-tagging**, as it was previously described in Subsection 2.1.1, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. Once again NLTK was used to extract this feature. In the possession of the first features, as mentioned before, each tuple containing these features are ran in the PoS tagger to extract the tag of each word of each sentence. Similar to the last topic, Pos bigrams will be used as features alongside the current word Pos. Pos bigrams can be used to determine which POS sequences can refer to an aspect term (DOHAIHA et al., 2018).
3. **Lemmatization** aims to remove inflectional endings only and to return the base or dictionary form of a word, which is known as *lemma*. For this study, WordNetLemmatizer from NLTK was used. The lemma for the word *deficiencies* is *deficiency*.
4. **Stemming** is similar to lemmatization, and aims to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form, ignoring stop words. SnowBallStemmer was used in this work.
5. **Superlative**, according to Penn Tree Bank project table of PoS tag, is when a term receives a tag of either JJS for Superlative Ajectives or RBS for Superlative Adverbs. Each token's PoS of the sentence is checked whether it has either of the superlative Pos tag, and a binary feature is saved in the column if a match happens.
6. **Comparative** receives either the PoS tag JJR or RBR for Comparative Adjectives and Adverbs, respectively. Analogous to the last feature, this feature receives a

binary value.

7. **Negative terms in the last four**. The meaning of a sentence can be changed, even turned to opposite, when a negative term is present. These terms can occur in words that are not close to the current term, taking the sentence *None of the customers enjoyed the food of the restaurant* have the modifier *None* that is distant from the verb *enjoyed*, that carries a positive meaning. The negative term inverted the meaning of the verb inside the sentence's context. The feature selected is a binary for whether the term existent four positions before each token is in the following vector: *['dont', 'never', 'no', 'nothing', 'nowhere', 'noone', 'none', 'not', 'hasnt', 'hadnt', 'cant', 'couldnt', 'shouldnt', 'wont', 'wouldnt', 'dont', 'doesnt', 'didnt', 'isnt', 'arent', 'aint', 'scarcely', 'cannot']*. For the terms that are in the first positions of the sentence, in other words, do not have four terms before itself, the feature is saved as 0.
8. **Positive Score** is the decimal score ranging from 0 to 1 given by SentiWordNet module *senti_synset*, which assigns to each synset of WordNet three sentiment scores: positivity, negativity, and objectivity. Due to the inability of the classifier algorithm to work with decimal values, the score is normalized to the range of 0 to 5. In this feature, each token receives a score of positivity according to *senti_synset*, where 0 is least positive and 5 is most positive.
9. **Negative Score**. Similarly to the previous feature, SentiWordNet is used, but this time is to determine the negativity score for each token. The same measures are used, where 0 means the token is least negative, and 5 most negative.
10. **Dependency Parsing Analysis**. The identification of the interdependence between terms of a sentence is primordial for communication, since syntactic relations provide meaning for linguistic context. Automatized methods for this task are of great necessity for the field. Spacy is an open source Python and Cython library that offers statistic modules of neural networks for many languages (Explosion AI, 2019). These systems are capable of identifying syntactic relations in sentence structure, as it is depicted in Figure 7. For this work, a new vector in sentence level was generated, where the words were not tokenized. The SpaCy module used in this task requires full sentences, punctuation and stop words included. Six new vectors were generated for each analysed sentence, each one of the vectors receives the term in its category. After this step, each token of the sentence is compared to the vectors contents. In case of a match, the feature corresponding to the matched vector received the value 1, if not, it received the value 0. Dependency parsing analysis was applied to identify the following dependencies:
    - **Noun subject**
    - **Direct object**
    - **Indirect object**

- **Copula**
- **Conjunction**
- **Coordinate Conjunction**

11. **WordNet Synsets** are a set of one or more synonyms that are similar in context without changing the meaning of the context. For this work, the library WordNet Synsets was used. The top 2 noun synsets of each word are used as features. The blanks are filled with "NULL" when the token does not have the needed number of synonyms.
12. **WordNet Hypernyms** As in the previous feature, WordNet Synsets library is used to find the parent and grandparent hypernyms of the current token. Hypernyms are words whose meaning includes a group of other words. When the needed number of hypernyms is not met, the blanks are filled with "NULL".
13. **Antonymy**, is when two words describe the opposite ideas of each other. WordNet Synsets was also used, and the first antonym of a said word is used. When this word has no antonym, "NULL" is saved in the column.
14. **Stop Words** are words that are most common in a language. In English, some examples of stop words are: 'are', 'which', 'the', and 'was'. Due to the inexistence of a single universal list of stop words used by all natural language processing tools, for this work the list provided by NLTK was used. If a word in a sentence is a stop word, it is marked as 1, if not, it is marked as 0.
15. **Frequent Aspect Term** A list of frequently occurring aspect terms in the training data, a term is frequent if it occurs at least 4 times. The feature is a binary if the term is frequent or not.
16. **Word bigram** Simple token bigrams using the current token, the previous and the next.
17. **PoS bigram** Analogous to the previous feature, PoS bigrams are used with the PoS tag for the current, previous and next token.
18. **Synset bigram** is a feature that differs from the regular bigram in the part that the order of the two adjacent synsets are not taken into consideration. This is to ensure a relatively high frequency of this type of feature, since not every word has an associated synset in the WordNet lexicon (BAAS et al., 2019).
19. **Synset PoS Bigram** is a combination between the top synset of each token and the token's PoS.
20. **Bigram output tag template** This feature is defined in the template file used by CRF++ classifier, defined as |output tag| x |output tag| x |all possible strings expanded with a macro|.

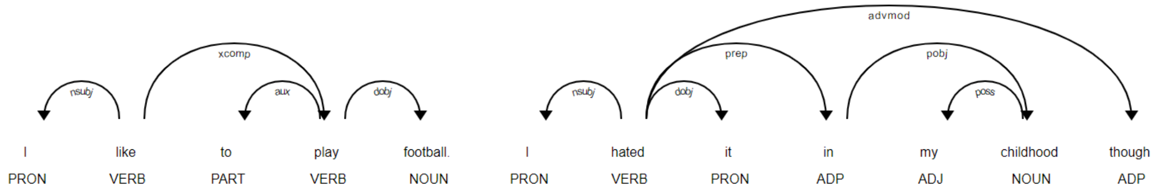
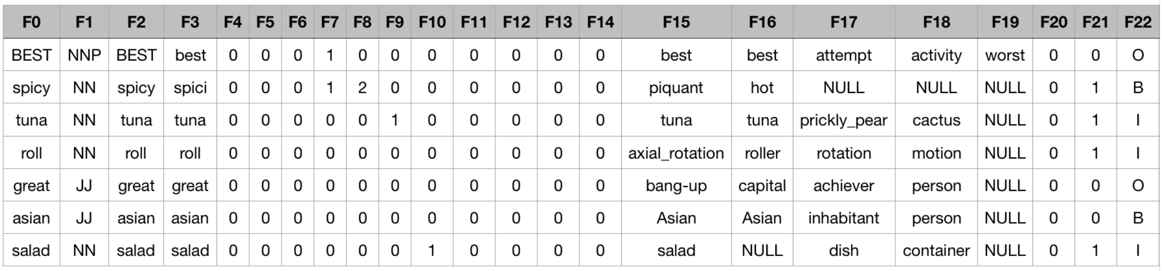


Figure 7 – Dependency parsing analysis model.

Source: Spacy



F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22
BEST	NNP	BEST	best	0	0	0	1	0	0	0	0	0	0	0	best	best	attempt	activity	worst	0	0	O
spicy	NN	spicy	spici	0	0	0	1	2	0	0	0	0	0	0	piquant	hot	NULL	NULL	NULL	0	1	B
tuna	NN	tuna	tuna	0	0	0	0	0	1	0	0	0	0	0	tuna	tuna	prickly_pear	cactus	NULL	0	1	I
roll	NN	roll	roll	0	0	0	0	0	0	0	0	0	0	0	axial_rotation	roller	rotation	motion	NULL	0	1	I
great	JJ	great	great	0	0	0	0	0	0	0	0	0	0	0	bang-up	capital	achiever	person	NULL	0	0	O
asian	JJ	asian	asian	0	0	0	0	0	0	0	0	0	0	0	Asian	Asian	inhabitant	person	NULL	0	0	B
salad	NN	salad	salad	0	0	0	0	0	0	1	0	0	0	0	salad	NULL	dish	container	NULL	0	1	I

Figure 8 – Example of file generated by the pipeline implemented in Python. Here are the 22 features and the IOB tag in the last column.

Source: The Author

4.1.2.2 Output File Generation

Once all three steps of the preprocessing and all features and labels are extracted, a matrix with this information is generated for each domain. In this document there are all sentences from the input corpora, segmented as tokens and the lines composed of the number of tokens in each sentence, and 23 columns, where each column is a feature and the last column is the IOB label. The sentences are separated by an empty line break, as it is required by CRF++, which will be explained in the Section 4.2.

After preprocessing of the sentences for restaurant and laptop reviews, and the extraction of features that can have relevant information, one output file is generated for each domain. Figure 8 shows an example of how the file is generated by the Python pipeline to be fed to the CRF++ algorithm. For each domain is created a training file to be used in the training phase and a test file to be used in the test phase with 23 columns, 22 of those being the previously described features and the last being the IOB tag column. The restaurant domain contains in the training file 43,929 lines and 11,699 lines in the test file. For the laptop domain, those numbers go up to 48,314 lines for the training file and 11,099 lines for the test file.

4.2 Aspect Term Extraction

As it is explained in Subsubsection 2.2.2.2, Aspect Term Extraction (ATE) aims to identify the aspects of given target entities in the domains. In this section it is explained how this task is achieved. Aspect extraction may be considered as a sequence labeling task because the product aspects occur at a sequence in a sentence (ZHANG; LIU, 2014). One of the state-of-the-art methods used for sequence labelling is Conditional Random Fields (CRF) (LAFFERTY; MCCALLUM; PEREIRA, 2001). This method takes as an input a sequence of tokens, calculates the probabilities of the various possible labeling options and chooses the one with the maximum probability.

Many tools are used to implement the CRF model, as it is described on the literature, some of the most popular ones are CRF++, CRFSuite, FlexCRFs, MALLET, RNNSharp, CRF-ADF, etc. For this work, CRF++ was chosen due to it's good results in the literature. It is written in C++ with TSL and is designed to be used with the command-line interface, easing its use. In addition to thins, CRF++ solves problems such as large scale numerical optimization, uses less memory both in training and testing, provides encoding/decoding in practical time and can perform n-best outputs.

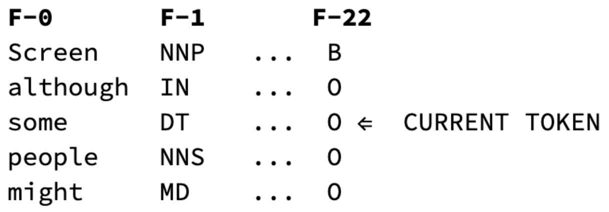
CRF++ is an open source project to implement the model CRF (Conditional Random Fields) that allows for segmentation and labelling of sequential data. It was developed to be used for generic goals, for this work it is used to extract the target expression of an aspect, or Aspect Target Extraction (ATE).

4.2.1 Experiment Configuration

The preprocessing previously described in Subsection 4.1.1 provides the data to be fed into CRF++ and creates data in a specific format so that the system can work. The training and testing files need to be in the same format, consisting of the number of tokens existing in the dataset, each line containing one token. After a sequence of tokens have become a sentence, and the sentences ends, a white space is necessary to separate it from the next sentence. For the features, columns separated by tabulation are used, each column receives a feature.

Along with these requirements, there is a certain semantic for the columns, meaning that each column (or set of columns) has its role in the files. The first column must always represent the token feature, in other words, the words and characters of the sentence in themselves. The second column must be the PoS tag for each token. The last column must represent a true answer tag which is going to be trained by CRF. CRF++ is designed to be of general use, so the number of columns is open, however the number of columns must be fixed through all tokens.

Also due to its generality, to use CRF++ it is necessary to specify the features



```
F-0       F-1       F-22
Screen    NNP   ...  B
although  IN    ...  O
some      DT    ...  O ⇐  CURRENT TOKEN
people    NNS   ...  O
might     MD    ...  O
```

Figure 9 – Example of input file denoting the current token

Template Macro	Expanded Feature
U00:%x[0,0]	some
U01:%x[0,1]	DT
U02:%x[-1,0]	although
U21:%x[0,0]/%x[1,0]	some/people
U22:%x[-1,0]/%x[0,0]	although/some
U23:%x[0,0]/%x[0,1]	some/DT
U24:%x[0,1]/%x[1,1]	DT/NNS

Source: The Author

Table 3 – Detail of the template file

templates in advance. These templates define which features are used in training and testing. Figure 9 and Table 3 shows an example of template used in this work, feature templates are descriptions for a CRF-based Algorithm, meaning that the template file sets up which features to use during a run of the CRF. Each line in the file denotes one template, each template denotes a special macro *%[row,col]*, which will be used to specify a token in the input data. The variable *row* specifies the relative position from the current focusing token and *col* specifies the absolute position of the column.

4.2.2 Training

After the preprocessing, labelling and feature extraction where the training and test files for each domain has gone through the same processes, the next phase is the training of the model. This process is done by using CRF++ command-line interface, where the command requires the use of the template file and the training input file in order to generate a trained model. These steps are done once for each domain, generating a model for each. The models contain all functions necessary to the classifier to infer the results on the test datasets.

4.2.3 Classification

The test phase, after the training is done and the models are generated, takes the test data previously prepared and formatted in the same style as the training data and applies the model generated by the training phase. Each test dataset contain 23 columns where 22 are the previously engineered features and the last is the label tag. There are 800 sentences in the test files, totaling approximately 11,000 tokens. Neither of the two domains contain repeated sentences on their training and test datasets, therefore cross-validation was not necessary. The methodology used in this work is the one suggested by SemEval 2014 in order to validate the results.

In testing phase, the files loaded are the test file and model file for each domain, the template file is not necessary anymore due to the model file having the same information as the template file. As mentioned before, the test files must be in the same format as the training files. Both datasets must contain the same number of columns, in this case, 23 columns, where the last is the classification tag that uses IOB notation. In CRF++ the last column is not suppressed and the application ignores it when the prediction phase is being done.

The classifier generates an output file with the same format and data as the input file containing classifier prediction column. By comparing the now two last columns (IOB and CRF prediction), the result can be analyzed, meaning that equal values in the two columns mean that the classifier got the prediction right, and a different value means that the classifier got it wrong. Chapter 5 discusses the results obtained in this work. It was necessary the use of an external tool to extract the results, since CRF++ does not provide the performance metrics in itself. *Scikit learn* [2] module *Sklearn metrics* [3] was used to measure the results, together with Python libraries *pandas* [4] and *numpy* [5], the metrics and tools are detailed in the following chapter

[2] https://scikit-learn.org/

[3] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

[4] https://pandas.pydata.org/

[5] https://numpy.org/

5 Experimental Evaluation: Results and Discussion

This chapter presents and analyzes the results yielded by the proposed solution to the problem of Aspect Term Extraction, providing a detailed explanation of the experimental setup, datasets and annotation schema, as well as, the discussion of the results. The experiments reported in this chapter were organized as experimental questions (EQ) that are presented in Section 5.2. At the end of the chapter we try to answer the raised experimental questions

In what follows, the datasets and their annotation schemas are described first, then the evaluations metrics, experimental protocol are presented next. In order to have a fair comparison with similar work, we adopted the same experimental protocol (datasets, dataset division for training/testing, and evaluation metrics).

5.1 Experimental Setup

5.1.1 Datasets

For the evaluation of the proposed system, two datasets composed by reviews of restaurants and laptops were used as the input data. These datasets were proposed by the SemEval 2014 Task 4 - Aspect-Based Sentiment Analysis in *xml* format. Figure 10 shows an example of an annotated sentence in the dataset.

Some statistics of both training and test data are provided in Table 4. The restaurants training data consists of 3041 English sentences and is a subset of the dataset from Ganu, Elhadad and Marian (2009), which included annotations for coarse aspect categories and overall sentence polarities. Aspect terms annotations occurring in the sentences, aspect

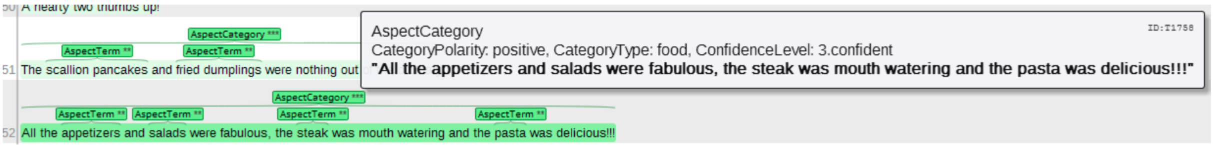
```
<sentence id="777">
        <text>From the appetizers we ate, the dim sum and other variety of foods, it was impossible to criticize the food.</text>
        <aspectTerms>
                <aspectTerm term="appetizers" polarity="positive" from="9" to="19"/>
                <aspectTerm term="dim sum" polarity="positive" from="32" to="39"/>
                <aspectTerm term="foods" polarity="positive" from="61" to="66"/>
                <aspectTerm term="food" polarity="positive" from="103" to="107"/>
        </aspectTerms>
        <aspectCategories>
                <aspectCategory category="food" polarity="positive"/>
        </aspectCategories>
</sentence>
```

Figure 10 – An XML Snippet showing an annotated sentence as it is in the dataset, where the sentence is inside the *<text>* tag, and each aspect term is described by the *<aspectTerm>* tags list.

Domain	Train	Test	Total	Positive	Negative
Restaurants	3041	800	3841	2892	1001
Laptops	3045	800	3845	1328	994
Total	6806	1600	7686	4220	1995

Source: Pontiki et al. (2014)

Table 4 – Sizes (sentences) of the datasets.



50 A nearly two thumbs up!

51 The scallion pancakes and fried dumplings were nothing out

52 All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!

AspectCategory
CategoryPolarity: positive, CategoryType: food, ConfidenceLevel: 3.confident
"All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!"

Figure 11 – A Sentence in BRAT Tool

Source: Pontiki et al. (2014)

term polarities, and aspect category polarities were added by the annotators. Additional restaurant reviews were collected and annotated from scratch in the same manner and used as test data, a total of 800 sentences. The laptops dataset is made up of 3845 English sentences extracted from laptop customer reviews. Annotators tagged the aspect terms and their polarities, the train data has 3045 sentences and the test data has 800 sentences.

5.1.1.1 Annotation Schema

The annotation task provides two types of information: aspect terms which is single or multiword terms naming particular aspects of the target entity; and aspect term polarities, that each aspect term is assigned to: positive, negative, conflict (both positive and negative sentiment), and neutral (neither positive nor negative sentiment). For the restaurant dataset, two annotation layers are present: aspect category, in which the annotator's task is to identify the aspect categories discussed in a sentence given the following five aspect categories: food, service, price, ambience, and anecdotes/miscellaneous; and aspect category polarity, which means that each aspect category discussed by a particular sentence has to be assigned one polarity. The annotation process was made using *BRAT* [1], a web-based annotation tool, which was configured appropriately for the needs of the ABSA task. Figure 11 shows an annotated sentence in BRAT, as viewed by the annotators. The annotation process took part in two stages described as follows:

1. **Aspect Terms and Polarities:** Consisting in the task of tagging all the single or multiword terms that named aspects of the target entity. Two annotators worked in this process, each one working with two subsets of 300 sentences from each dataset, and one reviewing the work of the other. The disagreements between the two annotators happened on borderline cases, being solved with the help of a third annotators. Most disagreements happened in the following three types

[1] https://brat.nlplab.org/

a) **Polarity Ambiguity:** The opinion of the reviewer was unclear to the annotators due to the lack of context.

b) **Multi-word aspect term boundaries:** There were disagreement in where the boundaries of multi-word aspect terms were when they appeared in conjunctions or disjunctions.

c) **Aspect term vs. reference to target entity:** Noun or noun phrases may refer to the entity as a whole or be used as an aspect. To help solve this problem, a broader context was needed.

2. **Aspect categories and polarities:** The objective of task is to tag each aspect appearing in each sentence with one of the five aspect categories, they are FOOD, SERVICE, PRICE, AMBIENCE and ANECDOTES/MISCELLANEOUS. Most disagreements were in the addition of missing aspect category. The same annotator validated the existing polarity labels in each aspect category annotation.

These tasks are performed with the dataset in English due to the shared task being proposed by SemEval 2014 in the English language. In order to apply these tasks in datasets consisting of different languages (e.g. Portuguese), it would be necessary a group of annotators with a reasonable level of familiarity to the chosen language, better if native speakers or people with a large knowledge in the different areas of study in linguistics. This is necessary to perform this manual task of annotation and provide a reliable annotated dataset.

5.1.2 Evaluation Metrics

To evaluate the performance of the proposed model, Precision (P), Recall (R) and F-measure (*F1*) were used. Precision (Equation 5.1) denotes the proportion of Predicted Positive cases that are correctly Real Positives. Conversely, Recall (Equation 5.2) is the proportion of Real Positive cases that are correctly Predicted Positive. This measures the Coverage of the Real Positive cases by the +P (Predicted Positive) rule (POWERS, 2011). F1 (Equation 5.3) relates precision and recall by an harmonic mean between those measures. With the use of F1, it is possible to retrieve the classifier's performance by using only one indicator, since it is a mean of Precision and Recall, it provides a more exact vision of the classifier's efficiency.

Other metrics such as accuracy and ROC were not used due to the fairness when comparing with similar work.

$$P = \frac{TP}{TP + FP} \tag{5.1}$$

$$R = \frac{TP}{TP + FN} \tag{5.2}$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5.3}$$

5.2 Experimental Questions

5.2.1 Running Time Performance

The first experimental question is what is the proportion of time spent to each phase in the proposed method. Taking all the steps into consideration, from preprocessing to result retrieving, the pipeline takes 28 minutes and 24 seconds for the restaurant domain and 35 minutes and 59 seconds for the laptop domain, by running in a Mac OS Catalina, with a 2,7 GHz Intel Core i5 Dual-Core Processor and an 8 GB 1867 MHz DDR3 RAM memory. This happens mostly because of the time it takes to preprocess the data. As it is explained in Subsection 4.1.1, the data is retrieved in *XML* format, containing only the sentences and the annotations, and it needs to be setup into the format accepted by CRF++, along with the feature writing. This process takes the longest time in the pipeline because it needs to be done by running through each word in each sentence, and as the training dataset is considerably larger than the test dataset (3041 sentences for restaurants and 3045 sentences for laptops training file versus 800 sentences for both test files), the time needed to preprocess the files is enormous. The positive is that the preprocessing step is only necessary to occur once, meaning that once the dataset is in the right format, training, testing and result retrieving can be done as many times needed.

Figure 12 details the time needed for each step of the pipeline for both datasets.

5.2.2 Study on feature importance - Ablation Study

The study on feature importance was done by performing an ablation test with the selected features, it responds the question of what are the impact of the proposed feature set on the system performance. The baseline result is the one with all the features on the template, as described in Subsubsection 4.1.2.1, then a series of tests is done by removing one or more features to observe how the system behaves.

The first test, as mentioned, is done with all the features, those include the twenty-two feature columns described in Subsubsection 4.1.2.1, word-level brigrams and template bigrams. For the second test, only the word-level bigrams are used, and there is a slight diminish on the result. The subsequent tests are done as follows:

- Test with only the base features.
- Test with only one feature at a time.

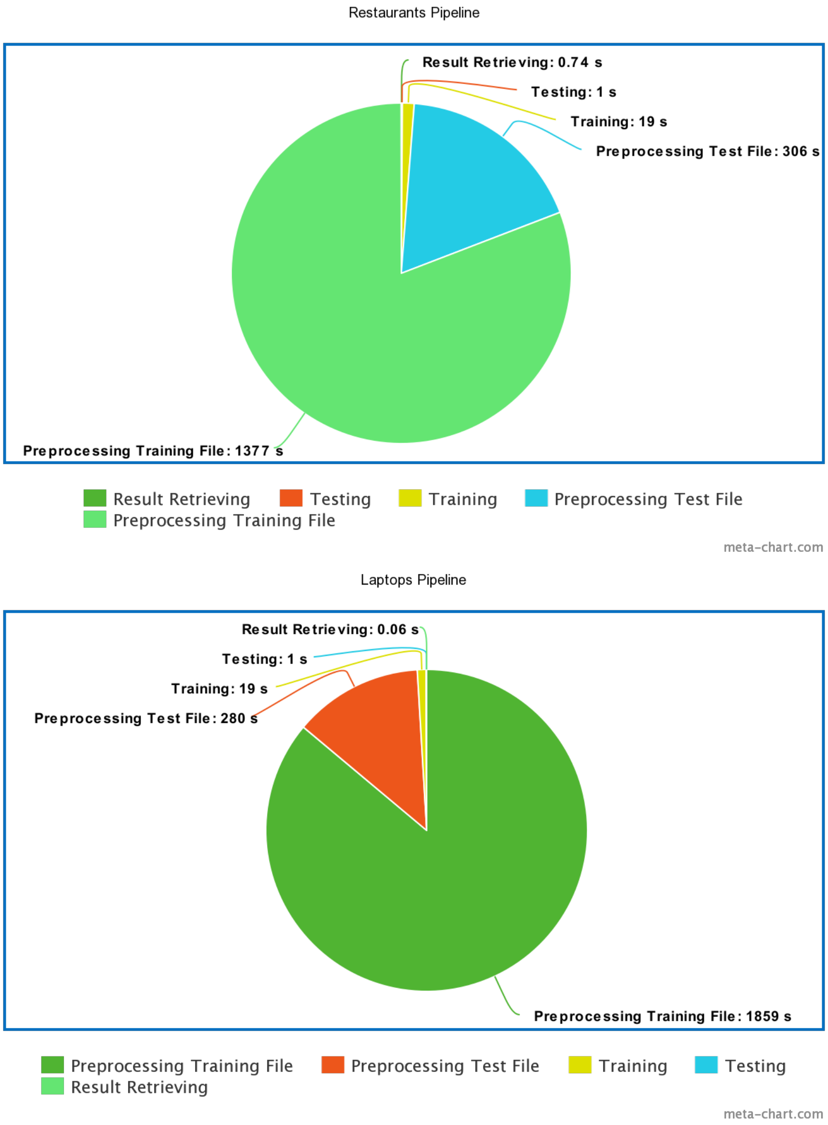


Figure 12 – Visualization of the time for each step of the pipeline in the restaurants domain

Source: The author

- Test with all the base features except one feature at a time.

This study yielded results that made possible to analyse how the features contributed to the performance of the system, showing which features helped or hindered the results.

By analysing the results, it is easy to see which features presented a larger positive contribution. If the first two tests are to be compared, the test containing all the base features, plus word-level bigrams and bigram templates yields a F-1 score of 76.68% for laptops and 79.42% for restaurants, while the test with all the base features and only word-level bigrams yielded 73.73% for laptops and 76.89% for restaurants. An almost four percent difference between results shows that, when dealing with term extraction, bigrams templates are very useful.

Test	Laptops	Restaurants
Base Features + Bigram Combinations + Bigram Template	76.68	79.42
Base Features + Word-level Bigram Combinations	73.73	76.89
without_isSuperlative	66.27	67.89
without_isComparative	65.90	67.88
without_lastFourNegative	65.73	67.87
without_antonym	65.82	67.86
without_negativeScore	65.87	67.82
without_copula	66.14	67.79
without_indirectObject	66.14	67.79
without_conjunction	65.15	67.77
without_coordinatingConjunction	65.90	67.72
Base Features	66.34	67.72
isComparative	60.59	59.93
directObject	60.66	59.74
antonym	60.98	59.68

Source: The Author

Table 5 – Main results of ablation tests done in the system

The analysis on the next tests shows the features that hindered the most the performance of the system. Test without features such as *is superlative*, *is comparative* and *last four negative*, yielded the best results after the results with bigrams.

Table 5 shows the most significant results in the ablation tests.

5.2.3 Comparative Evaluation

The measurements for the performance results of the system ADRL_UFRPE are made using Precision (P), Recall (R) and F-1 Score, as mentioned in Subsection 5.1.2. Deriving from that principle, in the restaurant domain, the system ADRL_UFRPE achieved 87.25% for precision, ranking third amongst the other systems. For recall, it reached 74.26%, staying in the seventh position, and for F-1 score, the system achieved 79.42%, also ranking seventh against the state-of-the-art systems.

The system achieved better results in the laptop domain, with 90.30% for Precision, ranking in first in comparison to the state-of-the-art systems. For recall, the system reached 69.01%, also ranking first. Lastly, for F1-score the system reached 76.68%, ranking first against the official SemEval 2014 results, standing behind only two systems developed by Xu et al. (2018) and Shu, Xu and Liu (2019), that also used the laptop dataset provided by SemEval 2014.

For the laptops domain, the system that achieved second best in precision was IHS_RD. (CHERNYSHEVICH, 2014) with 84.80 by applying different sets of features containing word features such as token, PoS and named entity; and using semantic label features. COMMIT (SCHOUTEN; FRASINCAR; JONG, 2014) scored 90.91 and ranked first in precision for the restaurants domain.

In the recall measure, for the laptops dataset, DLIREC (TOH; WANG, 2014) ranked second with a score of 67.13, and ranked first in the restaurants domain, scoring 82.72. These results can be attributed to the use of additional external resources (e.g. unlabeled data) to improve the extraction performance.

The second best F1 score was achieved by the IDS_RD. team with an *F1* score of 74.55% for laptops, which relied on Conditional Random Fields (CRF) with features extracted using named entity recognition, PoS tagging, parsing, and semantic analysis. They used additional reviews from Amazon and Epinions (without annotated terms) to learn the sentiment orientation of words and they trained their CRF on the union of the restaurant and laptop training data that we provided; the same trained CRF classifier was then used in both domains.

The best score for restaurants was achieved by DLIREC which also uses a CRF with PoS, dependency tree based features, and features derived from the aspect terms of the training data and clusters created from additional reviews from YELP and Amazon, scoring 84.01%.

When compared to the baseline on Aspect Term Extraction task proposed by SemEval 2014, the proposed solution is seen to have performed significantly better than the baseline in both datasets. Table 6 shows the comparison. These results are taken from the experiment with the best feature set in the study on feature importance. Pontiki et al. (2014) defined the sub-task baseline, in order to provide the ABSA participants with a mechanism to test their own purpose-built systems against and to demonstrate what can be achieved using simple methods. Each ABSA sub-task received its own baseline method. For Aspect Term Extraction, the baseline method was to build a collection of all aspect terms present in the training data, then identify aspect terms in the test data and lastly match these aspect terms with the ones collected from the previously built collection. In summary, this sole purpose of this method is to identify aspect terms that were previously tagged in the training data.

By analysing the results, it is possible to see that the feature set chosen to run the experiments was a differential in improving the results of the proposed system. As analyzed before, bigrams have shown the best results, and this system, in comparison to the other state-of-the-art system has done better due to the use of bigrams together with the most common word features. It can be seen that the proposed method achieved better results in the laptop domain, this can be attributed to the fact that in the restaurant set, many words appear only once (e.g., dishes, ingredients), and when words do not appear in the training set, no co-occurrence with any category can be recorded.

Table 7 brings the results achieved by the system ADRL_UFRPE, showing the

Laptops		**Restaurants**	
Team	***F1***	**Team**	***F1***
ADRL_UFRPE	76.68	ADRL_UFRPE	79.42
Baseline	35.64	Baseline	47.15

Source: The Author

Table 6 – Comparison between proposed solution and the baseline

comparison between the proposed method and systems that used the datasets provided by the Aspect-Based Sentiment Analysis task of SemEval 2014, which attracted 24 teams for the laptops dataset and 24 teams for the restaurants dataset. The table shows the results achieved for Precision, Recall and F-1 and orders the list by each result.

The proposed system can be compared to other systems based on CRF as a valid solution to the Aspect Term Extraction task, by outperforming all other submissions for laptops and staying in the top ten against 27 other submissions for the restaurants domain and staying well above baseline. When compared to SVM and other classifiers, the solution proposed in CRF can be seen also as a competitive proposal, due to the close proximity on performance, taking into consideration that on the state of art, other solutions proposed SVM and other classification algorithms to solve different tasks achieving similar results.

Laptops						Restaurants					
Team	**Precision**	**Team**	**Recall**	**Team**	**F1**	**Team**	**Precision**	**Team**	**Recall**	**Team**	**F1**
ADRL	**90.30**	**ADRL**	**69.01**	Ctrl-CNN (2019) *	82.73	COMMIT.	90.91	DLIREC	82.72	DLIREC	84.01
IHS_RD.	84.80	DLIREC	67.13	DE-CNN (2018) **	81.59	NILCUSP	87.73	XRCE	81.83	XRCE	83.98
COMMIT.	83.62	IHS_RD.	66.51	**ADRL**	**76.68**	**ADRL**	**87.26**	UNITOR	78.66	NRC-Can.	80.18
NILCUSP	83.62	UFAL	66.21	IHS_RD.	74.55	SeemGo	86.62	NRC-Canada	76.37	UNITOR	80.09
SeemGo	83.06	XRCE	64.98	DLIREC	73.78	XRCE	86.25	UWB	76.28	UNITOR	79.96
DLIREC	82.52	DLIREC	63.30	DLIREC	70.40	IHS_RD.	86.07	ECNU	74.69	IHS_RD.	79.62
SAP_RI	82.25	IIT_Patan	62.84	NRC-Can.	68.56	DLIREC	85.35	**ADRL**	**74.26**	**ADRL**	**79.42**
DLIREC	81.90	UNITOR	61.62	UNITOR	67.95	NRC-Can.	84.41	IHS_RD.	74.07	UWB	79.35
ECNU	81.03	NRC-Can.	60.70	XRCE	67.24	SAP_RI	83.69	SAP_RI	72.84	SeemGo	78.61
DLIREC	79.31	SNAP	60.40	SAP_RI	66.60	UWB	82.70	SINAI	72.49	DLIREC	78.34
NRC-Can.	78.77	SINAI	57.65	IITP	66.55	UNITOR	82.45	UFAL	72.49	ECNU	78.24
UNITOR	77.41	UNITOR	57.65	UNITOR	66.08	ECNU	82.15	IIT_Patan	72.13	SAP_RI	77.88
UWB	77.33	SAP_RI	55.96	SeemGo	65.99	JU_CSE.	81.85	SeemGo	71.96	UWB	76.23
lsis_lif	76.02	ECNU	55.50	ECNU	65.88	lsis_lif	81.20	Blinov	71.87	IITP	74.94
UNITOR	75.75	SeemGo	54.74	SNAP	62.40	USF	78.27	UWB	70.28	DMIS	72.73
USF	75.43	DMIS	51.38	DMIS	60.59	DMIS	78.19	EBDG	69.22	JU_CSE.	72.34
JU_CSE.	74.42	UWB	49.54	UWB	60.39	IIT_Patan	77.98	DMIS	67.99	Blinov	71.21
DMIS	73.85	JU_CSE.	49.39	JU_CSE.	59.37	UBham	77.95	JU_CSE.	64.81	lsis_lif	71.09
IIT_Patan	70.74	Blinov	48.93	lsis_lif	56.97	Blinov	70.56	USF	64.46	USF	70.69
XRCE	69.67	lsis_lif	45.57	USF	52.58	EBDG	69.35	V3	64.11	EBDG	69.28
EBDG	67.59	V3	42.51	Blinov	52.07	SINAI	59.61	lsis_lif	63.23	UBham	68.63
SNAP	64.54	USF	40.37	UFAL	48.98	V3	57.15	UBham	61.64	UBham	68.51
UBham	60.38	UBham	39.14	UBham	47.49	SNAP	57.14	iTac	39.59	SINAI	65.41
Blinov	55.65	EBDG	29.97	UBham	47.26	UFAL	49.58	SNAP	39.15	V3	60.43
UFAL	38.87	iTac	24.77	SINAI	45.28	iTac	37.08	COMMIT.	38.80	UFAL	58.88
SINAI	37.29	COMMIT.	14.83	EBDG	41.52			NILCUSP	34.04	COMMIT.	54.38
V3	32.18	NILCUSP	14.83	V3	36.62					NILCUSP	49.04
iTac	23.14			COMMIT.	25.19					SNAP	46.46
UBham	0.60			NILCUSP	25.19					iTac	38.29
				iTac	23.92						

Source: The author, based on Pontiki et al. (2014)

Table 7 – Results for Aspect Term Extraction. **Leg:** The lines in bold (ADRL) represent the system developed in this work; * System published in 2019 that used SemEval 2014 dataset; ** System published in 2018 that used SemEval 2014 dataset.

6 Conclusion and Future Work

This chapter summarizes and discusses the contributions, shortcomings and open paths of exploration for further improvement of the present research work.

The development of this work provided an overview of the use of word features to extract aspect terms of annotated sentences in Aspect-Based Sentiment Analysis. To achieve the goals of this work, the main features for automatic aspect extraction were studied, an CRF classifier was applied, achieving satisfactory results in comparison to the baseline on the state of art in the field of Sentiment Analysis.

Some findings were made during the development of this work and its experiments, those are:

- The best individual features are word features such as Token and PoS tags.
- The use of template bigrams improves the overall results considerably.
- Features of little semantic value do not help on the results.

The main contributions that can be taken from the development of this work are:

- The elaboration of a pipeline for the preprocessing, feature engineering, training and testing;
- The production of two feature matrices for training and test in a CRF classification model;
- A comparative assessment in which the proposed method was better in the task of Aspect Extraction in Sentiment Analysis, yielding state of art results in the task when compared with other systems.

Limitations

- The current work was an evaluation on SemEval datasets containing 3041 sentences for restaurants reviews and 3045 for laptops in the training files, and 800 sentences for both test files, which is relatively small. In a real world scenario, much bigger datasets are needed in order to test scalability of the system.
- Due to time limitation, it was not possible to perform the task of Aspect Category Detection.

- The system developed in this work is available only with datasets in English, due to the availability of tools for languages other than English (e.g. Portuguese, French) being more scarce. Therefore, the limitations for implementing in other languages are of retrieving or developing natural language processing tools that produce the same analysis and results that were used by this work to apply in English texts.

Future Work

1. Conducting further experiments on more and bigger datasets than the ones employed in this work. For real case application scenarios, such a study is very important for investigating the feasibility of the proposed method for ABSA.
2. Extending the approach to perform Cross-Domain Sentiment Analysis, the task of training a classifier in a domain (e.g. restaurants) with labelled datasets and testing it in a different domain (e.g. cars) without labelled datasets and achieving a satisfactory level of result. This task is done by transferring the knowledge from label rich domain (source domain) to the label few domain (target domain) and aiming to extract domain invariant features, whose distribution in the source domain is close to that in the target domain.
3. Continue extending the study to make the system applicable to other tasks such as Aspect Category Extraction (SemEval 2014 and SemEval 2015). SemEval 2015 ABSA Task (SE-ABSA 15) built up on SE-ABSA 14 and consolidated its subtasks (aspect category extraction, aspect term extraction, polarity classification) into a principled unified framework.
4. Adapting the system to more languages starting with Portuguese and French. Starting with the task of retrieving annotated datasets in those languages and adapting the preprocessing step to extract the features for each language.
5. Improving feature engineering, concerning the feature engineering step in the proposed solution, a first improvement resides in attempting to find other and even better combination of features, including bigrams and trigrams that could allow better results. In such case, a study on both computation time and memory should also be taken into account.

Bibliography

BAAS, F. et al. Exploring lexico-semantic patterns for aspect-based sentiment analysis. In: ACM. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. [S.l.], 2019. p. 984–992. 22, 30, 34, 35, 41

BECKER, K.; TUMITAN, D. Introdução à mineração de opiniões: Conceitos, aplicações e desafios. *Simpósio brasileiro de banco de dados*, v. 75, 2013. 25

BOIY, E.; MOENS, M.-F. A machine learning approach to sentiment analysis in multilingual web texts. *Information Retrieval*, v. 12, n. 5, p. 526–558, Oct 2009. ISSN 1573-7659. Available at: <https://doi.org/10.1007/s10791-008-9070-z>. 25

CHAWLA, R. *Overview of Conditional Random Fields*. 2017. Last accessed 4 December 2019. Available at: <https://medium.com/ml2vec/overview-of-conditional-random-fields-68a2a20fa541>. 27

CHERNYSHEVICH, M. Ihs r&d belarus: Cross-domain extraction of product features using crf. In: *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. [S.l.: s.n.], 2014. p. 309–313. 31, 34, 51

CUNNINGHAM, H. Gate, a general architecture for text engineering. *Computers and the Humanities*, Springer, v. 36, n. 2, p. 223–254, 2002. 17

DING, X.; LIU, B.; YU, P. S. A holistic lexicon-based approach to opinion mining. In: ACM. *Proceedings of the 2008 international conference on web search and data mining*. [S.l.], 2008. p. 231–240. 25

DOHAIHA, H. H. et al. Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems with Applications*, Elsevier, 2018. 24, 39

Explosion AI. *spaCy · Industrial-strength Natural Language Processing in Python*. 2019. Last accessed 4 December 2019. Available at: <https://spacy.io/>. 40

GANU, G.; ELHADAD, N.; MARIAN, A. Beyond the stars: improving rating predictions using review text content. In: CITESEER. *WebDB*. [S.l.], 2009. v. 9, p. 1–6. 46

HU, M.; LIU, B. Mining and summarizing customer reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004. (KDD '04), p. 168–177. ISBN 1-58113-888-1. Available at: <http://doi.acm.org/10.1145/1014052.1014073>. 21, 23, 25

JIANG, L. et al. Target-dependent twitter sentiment classification. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (HLT '11), p. 151–160. ISBN 978-1-932432-87-9. Available at: <http://dl.acm.org/citation.cfm?id=2002472.2002492>. 25

JIVANI, A. G. et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, v. 2, n. 6, p. 1930–1938, 2011. 23

KAUER, A. U. Análise de sentimentos baseada em aspectos e atribuições de polaridade. 2016. 29, 34, 35

KOK, S. de et al. level aspect-based sentiment analysis using an ontology. In: ACM. *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. [S.l.], 2018. p. 315–322. 30, 34

KUMAR, A. et al. Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In: *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*. [S.l.: s.n.], 2016. p. 1129–1135. 33, 34

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 43

LANDWEHR, N.; HALL, M.; FRANK, E. Logistic model trees. *Machine learning*, Springer, v. 59, n. 1-2, p. 161–205, 2005. 30

LIU, B. *Sentiment analysis and opinion mining*. [s.n.], 2012. ISBN 9781608458844. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.244.9480&rep=rep1&type=pdf>. 13, 21, 22, 24, 25

LOPER, E.; BIRD, S. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002. 17, 39

MACHACEK, J. Butknot at semeval-2016 task 5: supervised machine learning with term substitution approach in aspect category detection. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. [S.l.: s.n.], 2016. p. 301–305. 29, 34

MANNING, C. et al. The stanford corenlp natural language processing toolkit. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. [S.l.: s.n.], 2014. p. 55–60. 17, 22

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014. 26

MOVAHEDI, S. et al. Aspect category detection via topic-attention network. *arXiv preprint arXiv:1901.01183*, 2019. 30, 34

NLTK Project. *Natural Language Toolkit — NLTK 3.4.5 documentation*. 2019. Last accessed 4 December 2019. Available at: <https://www.nltk.org/>. 39

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (EMNLP '02), p. 79–86. Available at: <https://doi.org/10.3115/1118693.1118704>. 22

PONTIKI, M. et al. SemEval-2014 task 4: Aspect based sentiment analysis. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, 2014. p. 27–35. Available at: <https://www.aclweb.org/anthology/S14-2004>. 24, 47, 52, 54

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Bioinfo Publications, 2011. 48

Python Software Foundation. *19.7. xml.etree.ElementTree — The ElementTree XML API*. 2019. Last accessed 4 December 2019. Available at: <https://docs.python.org/2/library/xml.etree.elementtree.html>. 37

RUDER, S.; GHAFFARI, P.; BRESLIN, J. G. Insight-1 at semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02748*, 2016. 33, 34

SAIAS, J. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. [S.l.], 2015. 32, 34

SANTORINI, B. *Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision, 2nd printing)*. Philadelphia, PA, USA, 1990. 19

SCHOUTEN, K.; FRASINCAR, F.; JONG, F. D. Commit-p1wp3: A co-occurrence based approach to aspect-level sentiment analysis. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. [S.l.: s.n.], 2014. p. 203–207. 51

SHU, L.; XU, H.; LIU, B. Controlled cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1905.06407*, 2019. 51

SUTTON, C.; MCCALLUM, A. et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., v. 4, n. 4, p. 267–373, 2012. 27

TOH, Z.; SU, J. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In: *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*. [S.l.: s.n.], 2016. p. 282–288. 33, 34

TOH, Z.; WANG, W. Dlirec: Aspect term extraction and term polarity classification system. In: *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. [S.l.: s.n.], 2014. p. 235–240. 32, 34, 35, 52

TURNEY, P. D. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ACL '02), p. 417–424. Available at: <https://doi.org/10.3115/1073083.1073153>. 22

XIA, H. et al. Sentiment analysis for online reviews using conditional random fields and support vector machines. *Electronic Commerce Research*, Springer, p. 1–18, 2019. 31, 34

XU, H. et al. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*, 2018. 51

ZHANG, L.; LIU, B. Aspect and entity extraction for opinion mining. In: *Data mining and knowledge discovery for big data*. [S.l.]: Springer, 2014. p. 1–40. 31, 43

ZHOU, X.; WAN, X.; XIAO, J. Representation learning for aspect category detection in online reviews. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. 29, 34