Utrecht University

DEPARTMENT OF INFORMATION AND COMPUTING SCIENCES

MSc. in Artificial Intelligence

Master of Science Thesis:

COMBINING LEXICON-BASED AND DEEP LEARNING-BASED METHODS FOR AUTOMATED EMOTION ANALYSIS OF NEWSPAPER ARTICLES IN DUTCH

Supervisors:
Prof. Dr. J-J.Ch. MEYER
Prof. Dr. M. MOORTGAT
Prof. Dr. D. THIERENS
Prof. Dr. K. ZERVANOU
G. SOLAKIDIS

Author:
L. LORATO

February 2017

Contents

1 Introduction

"They're against happiness, this is why they attack me."
–Matteo Renzi. Interview on La Stampa. 8/11/2015.

The role that emotions play in the political debate has been subject of more and more attention in the recent years. While traditional studies in political science had been trying to relate political choices of the electorate only at a rational or social level, recent studies have been showing how the emotional appeal can be a key factor in political elections [21]; it definitely seems as a much important factor on the aftermath of a political election where a fact-free rhetoric has had its way in an age where fact-checking is a possibility available to almost everyone; as some have stated, there seems to be a gap between rational and emotional democracy [10].
While emotional appeals are observables in most of the politics of the last century, the direct influence of emotions in the dynamics of political communication has been rising only in the recent years, following the influence of media and technology.
The dynamics of media, the core of political debate and communication, have been changing following the new developments in technology. Journalism has become interactive, participatory, multi-platform and multi-linear, producing a constant stream of data, analysis and comments [4]. Much of these changes are due to the more direct ways through which people access news nowadays: smartphones and social networks are set to become soon the primary gate of access for readers, surpassing newspapers websites and going always further away from their paper editions [5].
This change in the dynamics of access have also been accompanied by an over-abundance of news: many studies show that people's attention span is getting more and more short to make it possible to process the huge amount of informations one is presented with by simply scrolling a social network's news-feed [32].
These factors have changed the way in which journalists and politicians communicate: the need to gain attention on such an overcrowded and immediate platform makes it only necessary for them to be as direct as possible. Media have been switching their focus in news-reports more and more over the years: from parties to single politicians, from rational opinions to emotional ones [21].
It seems plain why a growing number of research in journalism and political science has been asking for a more thorough study on these dynamics, on how politicians and news use emotions in their communication and on the effects produced by it [5] [21].

The aim of this thesis project, carried through at Parabot Services in Amsterdam, has been that of developing a system that could automatically analyse news articles and understand when an emotion was being expressed, trying to identify as well who was the expresser of the emotion and who was the target.
In the following we start in section 2 with an overview on the role that emotions have had in the field of AI, we then continue to expose more clearly in section 3 the problem that we aim to resolve and then in section 4 its lexicon-based solution is presented; the deep-learning based solution is exposed in section 5.

2 Emotions in AI

The idea that emotions are manifestation of an irrational behaviour is definitely not new and mentions of it can be found already in ancient western philosophy. According to the stoic school, a *sage*, a person who had attained moral and intellectual perfection, was the person able to resist emotions, considered as false judgements, or, in the words of the stoic philosopher Arius Didymus, "excessive impulses which are disobedient to reason" [3].
In more modern times, it was Darwin himself in his book *The Expression of the Emotions in Man and Animals* [12] to consider emotional expressions as no more than *fossils*, remnants of behaviours from an evolutionary past that were no longer useful. Emotional expressions were thus considered as involuntary and indicative of our primitive origins [43].
It was only in the second half of the twentieth century with cognitive psychology and with advances in neuropsychological research that the importance, if not the necessity, of emotions in human lives began to be proven [43]. Studies on individuals with damaged frontal lobes, and consequently incapable of experiencing emotions, showed in fact that such individuals, instead of becoming *super-rational*, were actually unable to make sensible decisions [43]. To explain these observations, Damasio proposed in 1994 the *somatic marker hypothesis*, according to which thinking of possible decisions causes learned emotional reactions to indicate whether the outcome will be good or bad [11]; therefore people unable to experience emotions were also impaired in their ability to make intelligent decisions.
Thus, this growing amount of psychological and neurological evidence for the relevance of emotions for intelligent behaviour has led consequently to an increase in research in AI fields related to emotions. Besides, even if it would have not been the case that emotions were necessary for rational reasoning, emotions would have been important for AI for the simple reasons

that they are important for humans: a robot or virtual character that wants to be believable to a human being needs to show empathy and be able to recognise and respond to manifestations of emotions in coherent ways, and just as well be able to predict emotional responses [43].

The field in AI that is related to the study of emotions is broadly called Affective Computing, as introduced by Rosalind Picard [39], including both those studies that try to recognise emotional manifestations in human beings and their converse, i.e. those related to artificial agents displaying emotional behaviours themselves [43].
In the latter category fall those studies that are related to modelling believable facial (or other bodily) expressions for robots or virtual characters, like, among others, the work of the FACE team at the University of Pisa [31]. Moreover, recent research in agent technology has also tried to expand Beliefs-Desires-Intentions models to consider the influence of emotions in the decision-making processes as in the research carried at the Intelligent Systems group at Utrecht University [33], trying to create formal representation of cognitive psychological models of emotions.
For what regards the converse field, the applications of it are just as various as various are the manifestations of emotions in humans; there have been a number of studies monitoring neurological or physiological signals trying to identify patterns that correspond to specific emotions, applications of such approaches are usually linked to wearable-related applications, as in the study from Lisetti and others [26] where physiological signals like galvanic skin response, heart rate and temperature were mapped to emotions like sadness, anger, surprise and so on.
Other approaches, instead, have been trying to identify emotion expressions using more *exterior* features such as facial expression [46] or emotional tone of voice [24].

2.1 Emotion and Sentiment Analysis

In the latter macro-category falls also the recognition of emotional expression at a linguistic level, also called *Emotion Analysis*; while the first linguistic studies of emotional expressions are traceable to the years 70s and 80s [37], a big increase in interest in such applications came around the turn of the century, mostly because of the increase of available text on the web, in the field of Opinion Mining or *Sentiment Analysis*, a field that can be considered as a simplified study on emotional expressions where these are classified only according to their positive/negative valence.
In reality the correspondence between emotional expressions and opinion expressions is not so straight-forward; opinion sentences such as "The food was

good", are not necessarily emotion expressions (even if it might be argued that one can still derive that if X thinks that the food is good, X is happy about the food). In any case, whatever the definition of emotional sentence and opinion sentences, the techniques used in the two applications are mostly the same, and for this reason they will be treated as related in the following, cases where the differences become relevant will be noted.

Studies on sentiment analysis have been mostly conducted at three different levels: document level, sentence level and aspect level [27]. The task of a sentiment analysis at *document level* is to classify whether a whole opinion document expresses an overall positive or negative sentiment [38]. The classic example in this task is movie reviews, where in general a number of positive and negative opinions sum up to an overall evaluation of the movie. This also implies that the system in this case gives as assumed that the document expresses opinions on a single entity (usually called the *target*).

A finer level of detail of analysis is that at *sentence level*. In this task the analysis is to determine whether the sentence expresses a positive, negative or neutral opinion [27]. This task has gained more importance in the recent years especially thanks to the raise in popularity of Twitter, the microblogging website has in fact become one of the main platforms people are using to express their opinions about social events and products they use in their daily life [1]. In this task as well, the target is given as assumed: a typical approach in this task would be of querying all the tweets that are mentioning a given target and then give them a sentiment classification.

As mentioned, both the document level and sentence level do not discover what exactly people liked and did not like [27]. The *aspect level* analysis focuses on the opinion expression itself, considering an opinion as composed of a sentiment (positive or negative) and a target.

Its application is therefore a finer-graned application of the previous approaches. Take the example sentence *"The iPhone's call quality is good, but its battery life is short"*, while it is expressing an opinion about the iphone, there are two specific opinion expressions about two different aspects of the iphone, its *call quality* and its *battery life* [27].

While the approach in this thesis can be considered a very fine-grained one, it differs from an aspect level analysis for a number of reasons, the first of which is that the system should identify not only the target, but also the *expresser*, i.e. who is expressing the emotion. The necessity of this study is also due to the different domain of application: while in product review it is quite likely that the expresser will always be the author of the text, this won't be the case in newspaper articles, considering the fact that such articles are often supposed to just report second-hand reactions of other people, emotional expressions in the text cannot be assumed to be related to the

journalist.
The second difference is that the target-identification approach can be regarded as more general than the classic aspect-finding one, as this will be mostly focused with identifying features related to the main target of a review. In a news report there might be many different emotions and targets in a single article.

3 Problem Formulation

In the following, a more detailed explanation of the goal of this project is given.
Given a sentence in Dutch, the system should be able to:

a. Infer whether the sentence is or not an emotional expression.

b. For emotional expressions, identify who is the expresser and who is the target of the sentence.

One can consider the following sentences and their classification as an example.

1. "Kiezers waren boos op de PvdA."[1]
   Voters were angry at the PvdA.
   Expresser: Kiezers
   Emotion: Anger
   Target: op de PvdA

2. "Voorzitter Jean-Claude Juncker van de Europese Commissie is blij met de verkiezingsuitslag in Nederland."
   The president of the European Commision, Jean-Claude Juncker is happy with the electoral results in the Netherlands.
   Expresser: Voorzitter Jean-Claude Juncker van de Europese Commissie
   Emotion: Happiness
   Target: met de verkiezingsuitslag in Nederland

3. "De burgemeester van Rotterdam Ahmed Aboutaleb vindt dat de PvdA geen verkiezingen meer moet houden om een partijleider te kiezen."
   The mayor of Rotterdam Ahmed Aboutaleb thinks that the PvdA should

[1]This, and all the sentences that will be used in the rest of this thesis come from data gathered by the ParaBotS system

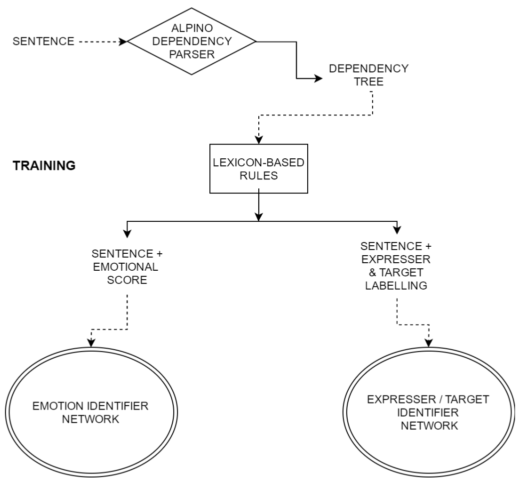

Figure 1: Diagram of the Emotion Analysis classifier, combining the lexicon-based approach with a deep learning based one.

> *not hold any elections any more to choose the leader of the party.*
> *Emotion*: Neutral

While a classical solution to Sentiment Analysis problems is dependent on an emotional lexicon, most of the recent studies in the field have been conducted using new developments in Deep Learning technology, showing numerous improvements over lexicon-based systems.
To the best of our knowledge, no studies had been yet made to apply deep learning approaches as such to an Emotion Analysis problem as ours and the goal of this project was also that of applying Deep Learning for the first time in such a field.
However, deep learning approaches usually need a lot of labelled data, data that is often labor-intensive and time-consuming to obtain through manual labelling. To overcome this difficulty, the approach in this project was that of combining a lexicon-based approach with a deep-learning based one by using the data labelled with the lexicon to train a deep learning architecture. Such a combination has been shown to be productive in improving the recall of the whole system in other studies such as the one from Zhang et al. in 2011 [22] for Sentiment Analysis of tweets, where a lexicon-based approach is combined with a Support Vector Machine classifier that learns from the results of the former.
The overall structure of the system is shown in Figure 1, in the following we start with the description of the lexicon-based classifier.

4 An enriched lexicon-based approach

The quite obvious start for finding emotion in text is starting from emotional words; people tend to be more expressive and descriptive in written text than what they tend to be in direct communication, given the lack of what would normally be clues of emotions, like the tone of voice and facial expression as mentioned earlier. Words like *blij (happy)* or *boos (angry)* are representations of what are sometimes called *direct emotion expressions*, and can be considered as strong clues of emotion. The first approach in this analysis is to develop a *lexicon* of emotional words, that can be used to relate a sentence containing such a a word to its corresponding emotion. In this approach therefore, sentence 1 would be classified as an anger sentence because it contains the word *boos*. However, as a number of studies in lexicon-based sentiment analysis have noted [27] [28], relying on only a lexicon for classifying sentences is far from sufficient considering that a sentence containing an emotion word is not necessarily an emotion expression.
There are a number of issues involved: an emotion word could appear in (syntactic) contexts that are not actually implying an emotion expression, like in negated sentences as *"Ik denk niet dat hij boos is" (I don't think that he is angry)* or in interrogative sentences and so on.
Moreover, an emotion word could have different meanings in different (syntactical) contexts, the example in Dutch is the emotional word *gelukkig*, whose meaning depend from the way in which it is used: in fact, while a sentence like *"Ik voel me gelukkig" (I am happy)* is a clear emotional sentence, the emotional content of a sentence like *"Daar zijn we gelukkig van af" (We're well rid of that)* is quite more debatable.
While the previous cases are dependent on syntactical clues (interrogative or negative sentence, word used as adjective or adverb and so on...) the situation gets even more complicated if one considers cases like sarcastic sentences or, even worse, sentences whose meaning is dependent on the context and on general knowledge. These last problems are major challenges in the whole Natural Language Processing (NLP) field and still far from being solved properly, however, being these out of the scope of this thesis, we won't deal with them further.

4.1 Lexicon

The lexicon used in this project is composed of few unambiguous words, a partial list of which can be seen in Table 1. The categorisation in Anger, Fear, Sadness, Disgust, Surprise or Happiness is due to Paul Ekman, who in

Anger	*boos, woest, woedend, wraakzuchtig, kwaad, haten, gefrusteerd, balen*
Fear	*bang, angstig, verward, gespannen, bibberig*
Sadness	*verdrietig, somber, gestresst, droevig, bedroefd*
Disgust	*walgen, braken, afschuwelijk, gruwelijk, vreselijk*
Surprise	*geschokt, verbaasd, verbijsterd, verwonderd, sprakeloos, ontzet, verrast*
Happiness	*blij, vrolijk, tevreden, gelukkig, verheugd, vreugdevol, dolblij, ontspannen, geamuseerd*

Table 1: Partial list of the emotional words that were used in the lexicon.

his cross-cultural study in 1992 argued for these as being the *basic emotions* [15]. More complex models have been proposed throughout the years in the literature, however Ekman's model has been by far the most used [45] and for this reason this was also the model used in this project. It is worth noting, however, that, the system explained in the remaining could be easily extended to alternative models.

Of course the degree to which a word expresses an emotion is in reality variable; some words, for example *balen (be fed up with)* can imply either disgust or anger, while others have in fact a different *magnitude*, e.g. a person saying to be *tevreden (satisfied)* is not as happy as a person saying to be *dolblij (overjoyed)*, just like a person saying to be *heel verrast (very happy)* is not as surprised as a person saying to be *een beetje verrast (a bit surprised)*. A multidimensional model of emotions is future work, for the rest of this paper it will be assumed that each word expresses one emotion only and all to a same degree.

Another aspect that is worth mentioning for the choice of the lexicon is that in this study the focus was on words that expressed emotional feelings, and not on words that caused emotional reactions.

4.2 Sentiment Analyisis in News

For what regards the second task, the identification of an Expresser and a Target for an emotion, a study that is related to ours is that conducted by Kim and Hovy in 2006 [23]. In their study they focus as well on news media text, trying to capture positive and negative opinion expressions, and, more importantly, trying to capture as well the expresser and the target in the sentence. As noted before, identifying such roles in a newspaper article is quite different from finding the features being evaluated in a product review, as there is no possibility to have a finite set of features as would be doable

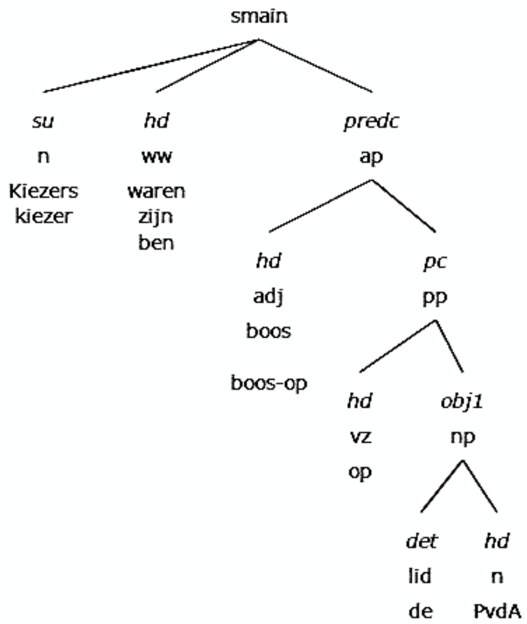

Figure 2: Alpino dependency tree for the sentence *Kiezers waren boos op de PvdA (Voters were angry at the PvdA).*

in the latter case.
To find the expresser and target in a news article, one can only rely on the structure of the sentence. The approach used by Kim and Hovy was that of using a *Semantic Role Labeling* tool [17]. Semantic Role Labeling is a NLP task for identifying roles such as Agent, Patient, Speaker and so on in a sentence. This is used as an intermediate step to identify the Target and Expresser among the candidate roles. For their semantic role modeling they base themselves on FrameNet, an online database that defines semantic frames for a number of words [2], and on other clues such as position of the word, voice of the sentence, phrase type and a parse tree.

4.3 Rules

Thus, both to enrich a simple lexicon approach and to identify the Expresser and the Target in the sentence one needs to use clues in the sentence structure. In the following, the approach used in this project is explained, where using Alpino, a dependency parser for Dutch, rules were defined for assuring the validity of emotion words and for the role labeling task.

4.3.1 Alpino

Alpino is a dependency parser for Dutch developed in the context of the PIONER project [8]. Its system creates dependency structures following the guidelines developed for the syntactic annotation for spoken Dutch in the project *Corpus Gesproken Nederlands (Corpus of Spoken Dutch)* [19]. Dependency trees make explicit the dependency relations between constituents in a sentence. An example of it can be seen in Figure 2.
In a dependency tree each non-terminal node has a list of daughters of which one is necessarily the *head-daughter* [8].[2] Moreover Alpino gives a number of labels to nodes in the tree, the main ones are the *dependency label* and the *category label*, which are also the labels appearing in Figure 2. Dependency labels give more details on the dependency relation between a node and its mother node, as mentioned before there must be always a node, among the daughters node that is head-daughter *(hd)*, other dependency labels are *subject (su)* or *direct object (obj1)*. Category labels instead are a union of *part-of-speech (POS)*, lexical tags and phrasal labels, such as *verb (ww)* or *nominal group (np)*; for a better overview on the annotation used in Alpino see [36].

As mentioned before, we need syntactical clues both to identify the Expresser and the Target and to enrich the lexicon-approach. For the latter aim, the problems that we tackle using Alpino dependency trees are the following:

1. Ensuring that an emotion word is in a syntactical context such that the sentence it is actually emotional,
2. Ensuring that an emotion word is used with the emotion-bearing meaning.

4.3.2 Handling Negation

For what regards the first problem, as mentioned there are various cases that make an emotional word invalid such as interrogative or negative sentences, but also cases where the emotional word appear in an *om-te* infinitive as in *"Simpele manieren om je elke dag gelukkig te voelen" (Simple ways to feel happy every day)* and other cases.
This problem, however, presents a number of complications that would require an ad-hoc study for a rule-based system, in the following the sketch of

[2] It should be noted that some literature makes a difference between *dependency trees* and *constituency trees*, considering the former as minimal tree representations, where there's a one-to-one relationship between nodes in the tree and words in the sentence. Alpino dependency structures have instead a one-to-one-or-more correspondence and should, if one wishes to be more precise, thus be considered a constituency parser.

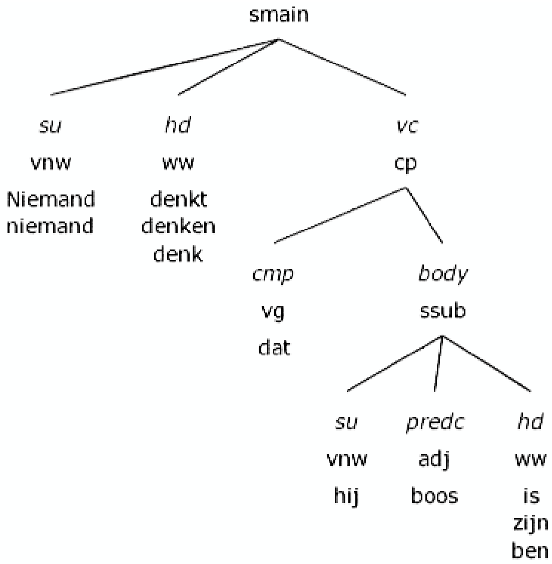

Figure 3: Alpino dependency tree for the non-emotional sentence *"Niemand denkt dat hij boos is" (Nobody thinks he is angry).*

a solution for the problem of negation is given.
We distinguish 3 cases in which an emotional word is negated:

1. The clause containing the emotion word is negated using adverbs such as *niet (not)* or *nooit (never)* as in *"Hij was niet tevreden" (He was not happy).*

2. The expresser or the target of the emotion expression is negated as in *"Niemand is boos" (Nobody is angry)* or *"Ik haat niemand" (I hate nobody).*

3. The clause containing the emotional word is a clause dependent on a negated main clause as in *"Niemand denkt dat hij boos is" (Nobody thinks he is angry).*

Needless to say, there are a number of exceptions to this rule: to mention a few, the construct *nog nooit* does not negate an emotion word as in *"Hij had nog nooit zo boos geweest" (He had never been this angry)* and there are cases where a negated main clause does not negate the dependent emotional clause as in *"Ik denk niet dat hij zo boos is op je" (I do not think he is so angry at you.).*
Not all the possible exceptions were handled, however, the system developed is able to recognise negation as expressed in the three cases above (and in

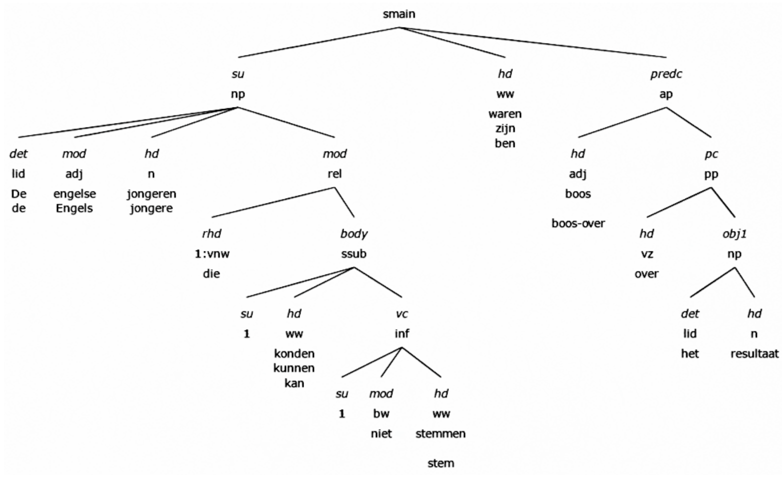

Figure 4: Alpino dependency tree for the emotional sentence *"De engelse jongeren die niet konden stemmen waren boos over het resultaat" (The english youngsters who could not vote were upset about the result).*

some irregular cases) by making searches in the dependency tree.
An example can be seen by considering the dependency tree in Figure 3; the system checks whether the emotional word is in a subordinate clause (with category label *ssub*), being that the case checks whether the main clause (labelled *smain*) is negated, that is, whether a negative adverb such as *niet* or *nooit* is part of the depth-one subtree or whether the subject of the main clause is a negative pronoun, being this the case for the sentence in Figure 3, it is marked as negative and the emotional word is ignored.
The scope for negative adverbs is limited at depth-one to avoid, especially for case 1, that sentences like the one in Figure 4 end up being marked as negative when they're actually not.
The checks for case 2 depend on the Target/Expresser identification module that will be explained later.

As mentioned, not all the syntactic phenomena that neutralize an emotional expression were dealt with in this project, however, the rules that were mentioned here briefly could be easily extended to solve the other cases.

4.3.3 Emotional Patterns

For what regards the second problem, we focus again on the example of the double use of *gelukkig* we had mentioned earlier. In Figure 5 the dependency trees are shown for both sentences; the interesting thing to notice is that the dependency labels of *gelukkig* in the two cases are different.

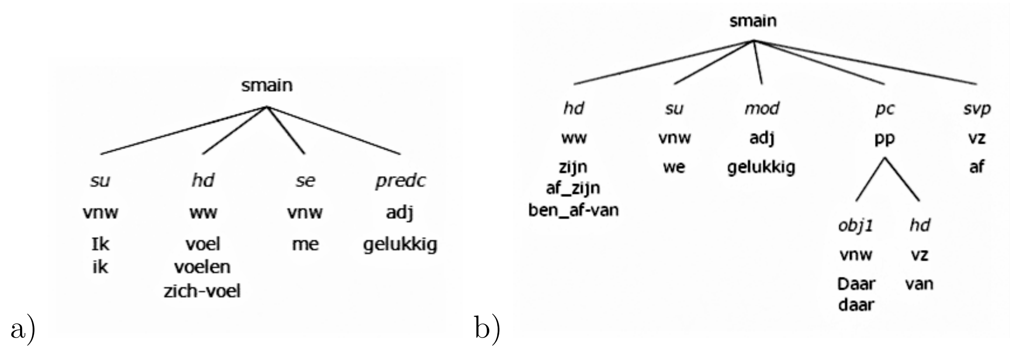

Figure 5: Alpino dependency trees for different uses of *gelukkig*. a) Tree for the emotional sentence *"Ik voel me gelukkig" (I am happy)*. b) Tree for the non-emotional sentence *"Daar zijn we gelukkig van af" (We're well rid of that)*.

While in the emotional sentence *gelukkig* has a *predc* label (which identifies a predicative complement), in the non-emotional one, its dependency label is *mod* (i.e. a general adverbial modifier). The key to make a lexicon-based approach more resistant to our first problem is thus to enrich the lexicon by adding to emotional words also the dependency label (and pattern) in which they appear; therefore, a sentence containing the word *gelukkig* would be classified as a happy sentence if *gelukkig* is in a subtree labelled with a *predc* dependency tag, not if it has a *mod* label.
It is worth noticing at this point that also *boos* in the tree in Figure 2 is in a *predc* subtree, and the same patterns are followed by other words like *bang (afraid)*, *verbaasd (surprised)*, *blij (happy)* and so on. Words expressing different emotions turn out to actually follow the same syntactical pattern for emotion expression and can therefore be gathered in a subclass that defines such rules. Even more interestingly, these subclasses are also used to define who is the expresser and who is the target in the sentence, as will be explained in the following section.

4.3.4 Finding Expresser and Target

As mentioned before, a key idea in Kim and Hovy's paper was that of using Semantic Role Labelling to identify the target and the expresser in an emotional sentence. Semantic Role Labeling however, can turn out to be an overcostly process and with a bigger scope of what would be needed for an emotion analysis system. Nonetheless, the important concept in this approach is that expressers and targets, while not classifiable with a semi-general approach as the one we mentioned for aspect-based analysis, can

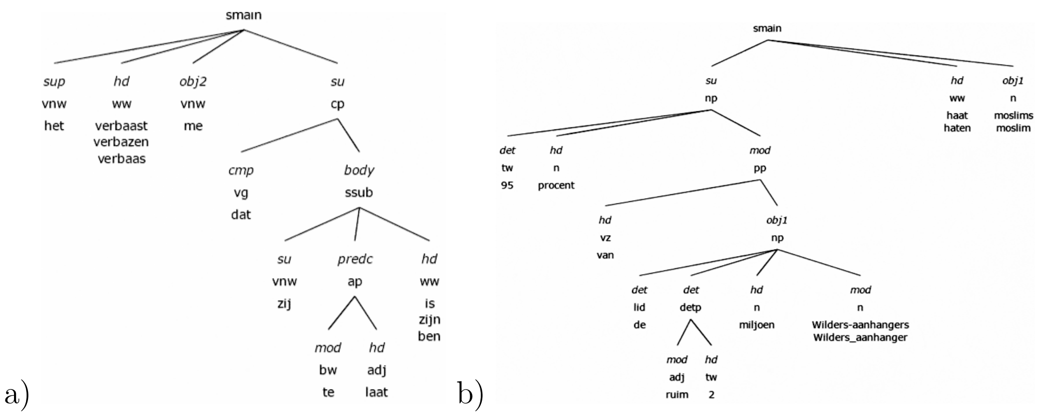

Figure 6: Different positions of expresser and target a) Tree for the emotional sentence *"Het verbaast me dat zij te laat is" (It surprises me that he is too late)*. b) Tree for the emotional sentence *"95 procent van de ruim 2 milioen Wilders-aanhangers haten moslims" (95% of the almost two millions supporters of Wilders hate Muslims)*.

be identified using the informations on how the emotion is being expressed. *Haten (to hate)* for example is a verb whose emotional content is directed from the subject of the sentence (the expresser) to its direct object (the target); verbazen (to surprise) instead is a verb whose emotional content has the opposite direction, that is, the expresser in such cases is the object and the target is the subject.
These cases can be identified using the dependecy trees as showed in Figure 6: in the sentence *"Het verbaast me dat zij te laat is" (It surprises me that she is too late)*, the expresser is *me*, i.e. the object in the sentence (with an *obj2* dependency label) and the target of the surprise expression is *"dat zij te laat is" (that she is too late)*, i.e. the subject of the sentence (*het (it)* in this case is labeled as *sup*, temporary subject, given that it refers directly to the content clause). Once again thus, the target and the expresser can be identified, given a certain emotion word, by simply making searches in the dependency tree.
As mentioned before, subclasses can be defined to describe the syntactical patterns in which an emotion word appears and also to define then where the expresser and the target will be located in the sentence. In this project we focused on four classes, that is:

- V1: The class of verbs where the subject is the expresser and the object (or an *"om te" infinitive*) is the target.

- V2: The class of verbs where the subject is the target and the object

A1	*boos, woest, bang, angstig, verbaasd, blij, gefrusteerd, vrolijk*
A2	*afschuwelijk, walgelijk, vreslijk*
V1	*haten, balen, walgen*
V2	*verbijsteren, verbazen*

Table 2: Partial list of the emotion words divided by syntactical subclasses.

is the expresser.

- A1: The class of adjectives where the subject is the expresser and the target is a *prepositional object* or a *verbal complement.*

- A2: The class of adjectives where the subject is the target and the expresser is the author of the sentence.

Of these four classes A1 was definitely the one to which most of the emotion words used in the lexicon belonged to. A partial list of the words belonging to each class is given in Table 2.

This way the system, using the dependency trees, is able to solve both the goals that were defined at the beginning of this project, that is:

a. Infer whether the sentence is or not an emotional expression.

b. For emotional expressions, identify who is the expresser and who is the target of the sentence.

And therefore, given the sentences of the first example, the system will provide exactly the answers that were expected.
Some of the limitations of this approach have been mentioned already and a full-fledged lexicon-based system would need many more rules and exceptions to be able to handle all the ambiguities inherent in a text-based analysis. Nonetheless this system provides us with a set of labelled sentences that can be used to train a deep learning architecture as will be explained in the next section.

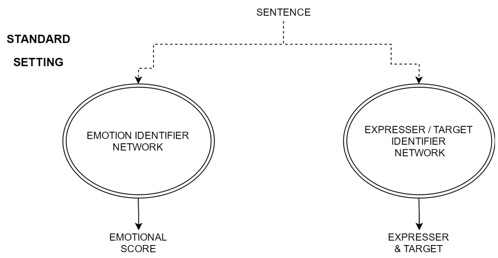

Figure 7: Diagram of the deep-learning architectures after training.

5 Neural architectures for Emotion Classification

The rise of interest in research for Deep Learning techniques in the recent years has influenced the field of Sentiment Analysis as well. A number of researches in this application showed promising results in overcoming the limitations of lexicon-based approaches and improving the adaptability of these classifiers to the ambiguities of natural language.
Non-rule-based approaches to Sentiment Analysis had been used since the research from Pang et al. in 2002 [38], where machine learning algorithms (such as *Naive Bayes* and *Support Vector Machine*) are used to build classifiers from manually annotated sentences. In most of the studies in this field therefore, researchers have been focusing on designing effective hand-crafted features to boost the performances of the classifiers. The problem with this approach is that engineering features is a labour-intensive process that does not always guarantee ideal performances [44].
The new approaches, instead, are *end-to-end* deep learning approaches and make thus possible to discover explanatory factors from the data without having to rely on feature engineering [7]. For the field of Sentiment Analysis, an effective way to learn features is to compose the representation of a sentence from the representations of the words it contains [40], whose representations (or *embeddings*) are thus learned using a deep network that computes a dense, low-dimensional and real-valued vector for each word [9] [35].
The deep learning architecture employed in this system to classify the emotion expressed in a sentence is therefore also an *end-to-end* approach, feeding word embeddings to a deep learning classifier, as explained in the following. The expresser/target classifier follows the same set-up and will be discussed in detail in the next section.

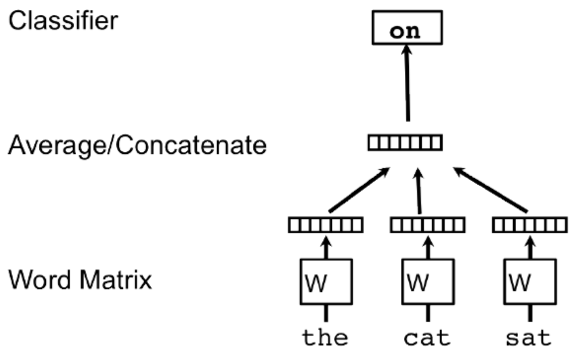

Figure 8: A framework for learning word vectors. The context of three words (*the, cat* and *sat*) is used to predict the fourth (*on*). [35]

5.1 Word Embeddings

Works like that of Pang et al. (2002) [38] used *bag-of-words* representations where each word is identified as a one-hot vector, meaning that the vector is as long as the size of the vocabulary, having only one dimension to 1, with all the others to 0.
The idea of deep-learning based word embeddings is that of reducing the sparsity (and the dimensionality problem) by using a distributed representation that could also substitute additional features, by also adding a notion of similarity between words to the same representation.
One of the most important works that implements word embeddings is that developed by Mikolov et al. (2013) [35] at Google, called *word2vec*. In word2vvec, every word is mapped to a unique vector, represented by a column in a matrix W. This representation is learned by using the concatenation or sum of the vectors as features to predict the next word in a sentence, as shown in Figure 8.
That is, given a sequence of training words $w_1, w_2, w_3, ..., w_T$, the objective of the word vector model is to maximise the average log probability

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, ..., w_{t+k})$$

The prediction task is done via a multiclass classifier, such as *softmax*, having thus

$$p(w_t | w_{t-k}, ..., w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

Where each of y_i is un-normalized log-probability for each output word i, computed as

$$y = b + Uh(w_{t-k}, ..., w_{t+k}; W)$$

with U, b as softmax parameters and h constructed by a concatenation or average of word vectors extracted from W [35].
After the training converges, words with similar meanings are mapped to a similar position in the vector space, so that, for example, as Mikolov notes, *"powerful"* and *"strong"* will be close to each other in the vector space, while *"powerful"* and *"Paris"* will be more distant. Also the difference between word vectors carries meaning, and they can be used to answer analogy questions by using simple vector algebra, for example, "*King*" - "*man*" + "*woman*" = "*Queen*" [35].
Because of these properties, and because of what discussed previously, word embeddings have been used in many tasks, such as machine translation [50], document understanding [48] and of course sentiment analysis and are becoming one of the most important techniques for deep-learning based studies in Natural Language Processing.

5.2 Deep Learning Architectures

A number of different architectures, mostly developed in 90s already, have been used in deep-learning techniques. While the *Convolutional Neural Network (CNN)*, a feed-forward network inspired by the organisation of the visual cortex, has definitely had wide application in the field of image and video recognition and some other NLP tasks, mostly at character-level, it has not been used much in Sentiment Analysis tasks, apart from some limited exceptions [14]. The problem with CNNs is that, while they allow to encode arbitrary large input in fixed size vectors that capture their most salient features , they sacrifice most of the structural informations, informations that can be crucial when dealing with sentences, where linguistic structure can give us important clues [18].
Two other architectures have instead proved effective for such tasks, *Recurrent neural networks* and *Recursive neural networks.*

5.2.1 Recursive Neural Networks

Recursive architectures are designed to take a structure as input, such as a dependency tree and apply recursively the set of weights over the structure. As shown in Figure 9 the classification of *"not very good"* is computed on the representation of the said phrase which is obtained recursively by first

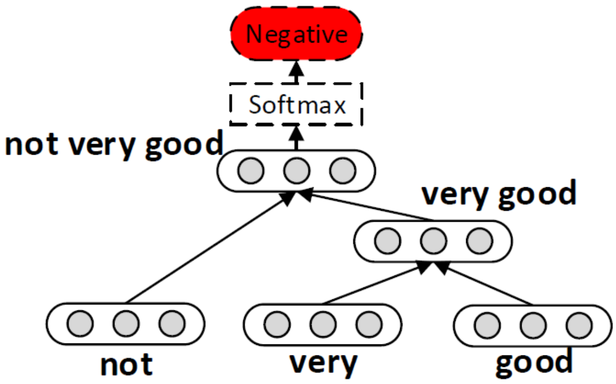

Figure 9: The composition process for "not very good" in a Recursive Neural Network. [13]

composing those of *"very"* and *"good"* and then those of *"not"* and *"very good"*, the composition of a new vector representation is computed as follow:

$$v = f\left(W\begin{bmatrix} v_l \\ v_r \end{bmatrix} + b\right)$$

where v_l, v_r are the vectors of its left and right child, W is the composition matrix and b is the bias vector.
This is the architecture used in the work by Socher et al. (2011, 2013) [40] [41], and, with a more elaborate mechanism of adaptive weights, by Dong et al. (2014) [13] for Sentiment Analysis.

5.2.2 Recurrent Neural Networks

The recursive architectures that we just described could actually be seen as a generalised recurrent neural network. The recurrent neural network (RNN), introduced first by Elman (1990) [16] is basically a network whose hidden state at time t is dependent not only on its input, but also on the hidden state from $t-1$, they allow representing arbitrarily sized structured input in a fixed-size vector but at the same time to pay attention to the structured properties of the input [18].
In the Elman-type RNN (Figure 10a), the output of the hidden layer h_t is computed from a non-linear transformation of the current input x_t and the previous hidden layer output h_{t-1}, i.e.:

$$h_t = f(Uh_{t-1} + Vx_t + b)$$

where f is the non-linear function (e.g. `sigmoid`) applied to the hidden units while U and V are weight matrices and b is the bias vector. Thus h_t can be

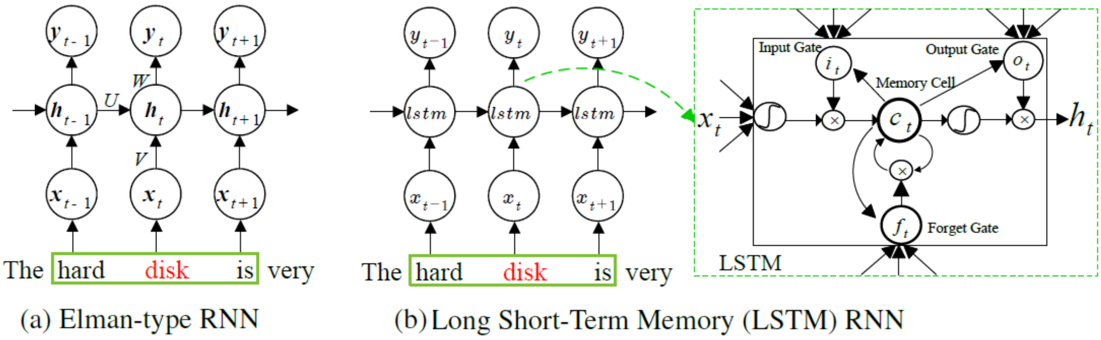

Figure 10: Elman-type and LSTM RNNs with an input layer, a hidden layer and an output layer. [29]

seen as evaluating the informations on the current input while taking into account the past states.
This RNN is usually trained using *stochastic gradient descent* with *backpropagation through time (BPTT)*, where errors (i.e. *gradients*) are propagated back through the edges over time [29]. BPTT however, suffers from the so-called *vanishing* and *exploding* gradient problem [6]. As the errors get propagated through time, they can become very small or very large and lead to undesired values in the weights, making thus the training fail. A solution to this problem it that of using a truncated BPTT [34] that restricts the backpropagation to a limited number of steps, usually 4 or 5. This solution however limits the RNN's capability of capturing long-range dependencies.
For this reason, usually more *sophisticated* RNN architectures are used, an example of this is the *Long-Short Term Memory (LSTM)* architecture, introduced by Hochreiter and Schmidhuber (1997) [20], it was designed to specifically model long term dependencies in RNNs. The hidden layer in a LSTM is constituted by special units, called *memory blocks*, as shown in Figure 10b, a memory block is composed of four elements:

- A memory cell c (i.e. a neuron) with a self-connection
- An input gate i to control the flow of input signal into the neuron
- An output gate o to control the effect of the neuron activation on other neurons
- A forget gate f to allow the neuron to adaptively reset its current state through self-connection

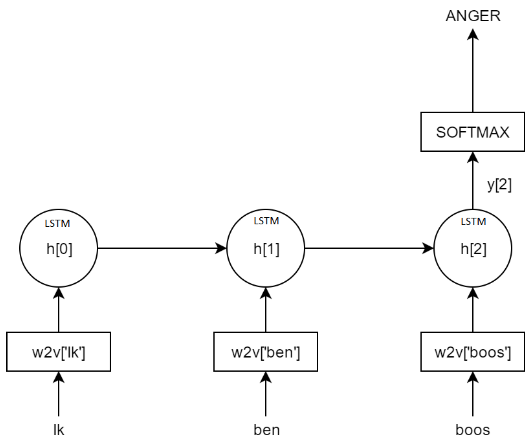

Figure 11: Diagram of the network architecture used for emotion classification.

whose equations are the following:

$$\begin{aligned}
i_t &= \sigma(U_i h_{t-1} + V_i x_t + C_i c_{t-1} + b_i) \\
f_t &= \sigma(U_f h_{t-1} + V_f x_t + C_f c_{t-1} + b_f) \\
c_t &= i_t \odot g(U_c h_{t-1} + V_c x_t + b_c) + f_t \odot c_{t-1} \\
o_t &= \sigma(U_o h_{t-1} + V_o x_t + C_o c_t + b_o) \\
h_t &= o_t \odot h(c_t)
\end{aligned}$$

where U, V and C are weight matrices associated with each gate, and b is the bias vector. The symbol $\odot$ denotes a element-wise vector product, σ is the `sigmoid` function and g and h are activation functions, typically `tanh` [29]. Because of its ability to model long-term dependencies, LSTM has been one of the most used architectures in the recent years for most of the deep-learning based technologies in natural language processing, and also in Sentiment Analysis there have been a number of works using LSTM RNNs [29] [47].

5.2.3 Setup

For the emotion classification task it was decided that the most appropriate architecture would have been the LSTM RNN. The design of a Recursive Neural Network dependent on Alpino trees, while being an interesting possibility, had a number of drawbacks, due to difficulties in the implementation because of the Alpino parser being a constituency tree[3] and speed limitations

[3]See footnote 2 at page 11

but also because of results in research did not show big improvements of such an architecture over LSTM ones [35] [47].
The architecture for the emotion classification task is shown in Figure 11. Each word is transformed into a word2vec vector and then passed to the LSTM, the hidden state of the last word in the sentence (since it stores relevant informations from the past) can be considered as a representation of the whole sentence and is thus used as input to the softmax layer for the classification between the 7 classes (Anger, Happiness, Sadness, Disgust, Fear, Surprise, Neutral).
The performances of the model will be discussed in Section 7.

6 Deep-learning based Expresser and Target identification

As already mentioned, identifying the expresser and the target in a sentence can be considered as a semantic role labelling task, and thus as a kind of *linguistic sequence labelling* task, just like part-of-speech tagging and named entity recognition. For these tasks as well, the recent advances in deep learning made it possible to develop end-to-end solutions that did not depend anymore on manually-engineered features.
A number of these systems have been designed, relying on a setup not too dissimilar from the one described in the previous section [49] [30] [25]. This architecture is also the one used for the task of identifying expresser and target, as showed in Figure 12, composed of a bi-directional LSTM combined with a CRF (*conditional random field*) layer on top, whose details will be described in the following. It is also worth noticing that, differently from the previous architecture, in this setup the classification is not done only on the last word of the sentence, but on each word, as would be expected after all, considering the task.
The architecture is designed then again as a classification task, where each word is assigned one of the following labels :

- EXP when part of the Expresser of the emotion,
- OPN when part of the emotion expression,
- TAR when part of the Target of the emotion,
- O when not part of any relevant component.

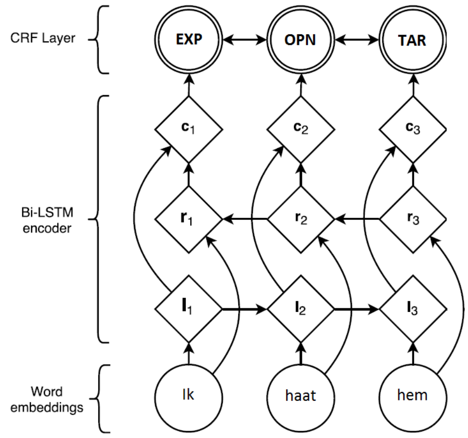

Figure 12: Diagram of the network architecture used for expresser and target identification. Word embeddings are given to a bidirectional LSTM. l_i represents the word i and its left context, r_i represents the word i and its right context. Concatenating these two vectors yields a representation of the word i in its context, c_i, this representation is then fed to the CRF layer that jointly assigns a label to each word [25].

6.1 Bi-directional LSTM

In the standard LSTM described in the previous section, each state receives informations only from the past, which is enough for a sentence classification task where the classification is done at the end of the sentence where thus all the relevant informations have been already processed.
It is not the case however for sequence labelling tasks where classification is done on each word and where thus informations from the future could be crucial. For example in the sentences *"Hij verbaast me" (He surprises me)* and *"Hij is boos" (He is angry)* it would be quite hard to establish with no informations from the future whether *hij* is the expresser or the target, not to mention cases in which it would just have to be considered as not part of any component.
The bi-directional LSTM is used thus just for cases like this, to incorporate long-term informations both from the future and from the past. A bi-directional LSTM is made of two independent layers of LSTM, as showed in Figure 12, they both receive the input word embedding at each time-step and one processes the sentence in forward-direction, one processes it backwards. The outputs are then combined and passed on to the upper layer.
It is worth noticing that, since the computation is done independently, during training, after backpropagating the error from the upper layer to the forward

and backward LSTM, two independent BPTT can be applied - one to each direction [29].

6.2 Conditional Random Fields

Beside considering informations from the future, for such a task it's also beneficial to take into account the correlations between labels in the neighbourhoods in order to assign jointly the best chain of labels for an emotional sentence [29]. It is true that ambiguities in our system are limited by considering only sentences where one emotional expression is present, thus cases like *"Ik haat iedereen die gelukkig is" (I hate everybody who is happy)* were not taken into account, however there are still cases where jointly labelling the sentence could be more effective.
For example, in sentences like *"Hij kwam thuis maar Mark was toen te boos" (He came home but Mark was too angry then)* it is hard to assign with an independent classification only the appropriate labelling of the Expresser (i.e. *hij (he)* would be classified as O only because of the information that *Mark* is classified as EXP). Conditional random fields serve exactly this purpose, they're formally defined as follows.
Given an input sentence $X = (x_1, x_2, ..., x_n)$, we consider P as the matrix of scores output produced by the bi-directional LSTM, P will be of size $n \times k$ where k is the number of distinct tags, and $P_{i,j}$ corresponds to the score for the $j^t h$ tag of the i^{th} word in a sentence. Thus, given a predictions on label assignments for the whole sentence $y = (y_1, y_2, ..., y_n)$, we define its joint score to be:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

where A is a matrix of transition scores such that $A_{i,j}$ represents the score of a transition from tag i to tag j. y_0 and y_n are the *start* and *end* placeholder tags of a sentence, that are added to the set of possible tags, thus A is a square matrix of size $k + 2$ [25]. A softmax over all possible tag sequences yields a probability for the sequence y:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\dot{y} \in Y_x} e^{s(X,\dot{y})}}.$$

During training, the log-probability of the correct sequence is maximised, that is:

$$log(p(y|X)) = s(X, y) - log\left(\sum_{\dot{y}\in Y_x} e^{s(X,\dot{y})}\right)$$

where Y_x represents all possible tag sequences for a sentence X [25].

7 Experiments

7.1 Experimental settings

Dataset: The data gathered at ParaBotS and pre-classified using the rule-based system discussed earlier contained around 80.000 sentences that were classified as emotional.
For the LSTM architecture for emotion classification (that for brevity we'll call from now on as *spotem*) the training data was filtered and augmented when necessary to have a more balanced distribution of data among the classes (circa 10.000 sentences per class), this means that data from classes such as *anger* or *happiness* was limited, while data from classes such as *surprise* and *disgust* was augmented using *data augmentation* techniques[4]. Beside the emotional sentences, also 30.000 neutral sentences were added to the training data (including the sentences containing emotional words but classified as neutral by the rule-based classifier).
For the bi-LSTM+CRF architecture for expresser and target labelling (that for brevity we'll call from now an as *exptar*) all the 80.000 sentences were used as training data, and no neutral sentences were added to the training set, as the network is supposed to label roles in a sentence that is emotional.

Word Embeddings, Spotem and Exptar: The word2vec embeddings were trained using 18 million sentences and with vectors of size 300. The trained word2vec dictionary contained 9.300 words (words that appeared less than 25 times were ignored).
Both in *spotem* and *exptar* the LSTM cells were of size 300 each. In both the networks, *dropout* was implemented between the input layer and the LSTM one. Dropout is a technique to avoid overfitting [42] that sets, at each training stage, individual nodes to 0 with probability p (in our case 0.5). This

[4] Sentences for the classes to be augmented were replicated by adding noise to their vector representation in order to obtain duplicates that were between 90% and 95% similar to the original.

way the network learns to not depend too much on certain *strong* patterns, which also leads to good generalisation performances [25].

7.2 Results and Discussion

It has already been noted how the lack of manually trained data and the dependency of the neural networks on a rule-based system with its limitations was not the ideal setup, given these assumptions however, the points of interest for the results of the networks were the following:

- whether the networks were able to learn effectively from the rule-based system,
- whether the networks were able to *generalise* over rule-based system by learning new words as emotional that were not present in the original lexicon.

Accuracy vs the rule-based system: Both the networks were able to learn from the training data with quite good results. The accuracy of *spotem* on a testing data of 8000 sentences was of 92%, and that of *exptar* was of 89.5% showing accuracies similar to other applications of such architectures as those mentioned earlier in part-of-speech tagging or Named Entity Recognition [30] [25], showing that the networks are actually able to learn from the rule-based system and imitate its behaviour, without any lexicon and any parsing tree structure.

Precision and Generalisation of the network: To understand better the behaviour of the model, we analysed its results on the text of a short-story of 3.900 sentences that were not part of the training. The performances of *spotem* and of the rule-based classifier were the following:

- Out of the 3.900 sentences, *spotem* classified 108 of them as emotional, the rule-based system classified 23 as emotional.
- Out of the 23 classifications made by the rule based system, 2 of them were wrong.
- Out of the 108 classifications made by *spotem* 42 were wrong, of these:
  - 36 were from wrong generalisations (i.e. wrong classifications on words that were not originally part of the lexicon).
  - 4 were from wrong classifications compared to the rule-based system.

  - 2 were from wrong classifications equal to the rule-based system.

- Of the correct classifications made by *spotem*, 45 were made by generalisation on words that were not present in the original lexicon but that do actually have emotional content.

- On the 66 correct classification made by *spotem*, *exptar* identified the expresser (and target) correctly 60 times[5].

It was not possible to estimate recall, given the amount of sentences, but the results are promising in showing that *spotem* is behaving as one would wish, being able to generalise and thus improve the recall of the rule-based system, even if in this case the generalisation seems a bit too noisy.

8 Conclusion and Future Work

In this thesis project a system was developed in order to perform emotion analysis and more specifically a) classify a sentence as expressing one of the 6 Elkman emotions or being neutral, and b) identify in an emotional sentence, the expresser and the target of the emotion.
The lack of manually annotated data for training a deep learning architecture led to the construction of rule-based classifier where a limited lexicon of strong emotional words was enriched with dependency-tree based rules. This system was used to then train two neural networks, a LSTM for emotion classification and a bi-LSTM+CRF for expresser and target labelling. Both the networks used word embeddings as input.
The performances of the networks were promising, despite the limits inherited from the rule-based classifier, and show that such techniques can definitely be effective in Emotion Analysis tasks. Future work should include, experimentation with manually annotated data, improvements on the generalisation capabilities of the network, dealing with multi-dimensional and multi-class classification (i.e. consider the emotional content of a word as being variable and also as being divided in multiple classes). Finally the framework should be also applied to English, trying to possibly create a unique, multi-lingual model.

[5] For this judgement we refer to an *overlap* accuracy, i.e. the labelling is considered as correct if the actual target or expresser are part of the labelling made.

References

[1] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." *Proceedings of the workshop on languages in social media. Association for Computational Linguistics*, 2011.

[2] Baker, Collin F., Charles J. Fillmore, and John B. Lowe. "The berkeley framenet project." *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*-Volume 1. Association for Computational Linguistics, 1998. APA

[3] Baltzly, Dirk. "Stoicism", *The Stanford Encyclopedia of Philosophy* (Spring 2014 Edition), Edward N. Zalta (ed.), URL = https://plato.stanford.edu/archives/spr2014/entries/stoicism/.

[4] Beckett, Charlie. "The value of networked journalism." (2010).

[5] Beckett, Charlie, and Mark Deuze. "On the Role of Emotion in the Future of Journalism." *Social Media+ Society* 2.3 (2016).

[6] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks* 5.2 (1994): 157-166.

[7] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013): 1798-1828.

[8] Bouma, Gosse, Gertjan Van Noord, and Robert Malouf. "Alpino: Wide-coverage computational analysis of Dutch." *Language and Computers* 37.1 (2001): 45-59.

[9] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." *Journal of Machine Learning Research* 12.Aug (2011): 2493-2537. APA

[10] Corner, John, and Dick Pels, eds. *Media and the restyling of politics: Consumerism, celebrity and cynicism*. Sage, 2003.

[11] Damasio, Antonio R. "Descartes' error: Emotion, rationality and the human brain." (1994).

[12] Darwin, Charles, Paul Ekman, and Phillip Prodger. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

[13] Dong, Li, et al. "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification." ACL (2). 2014.

[14] Dos Santos, Cícero Nogueira, and Maira Gatti. "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts." COLING. 2014.

[15] Ekman, Paul. "An argument for basic emotions." *Cognition & emotion* 6.3-4 (1992): 169-200.

[16] Elman, Jeffrey L. "Finding structure in time." *Cognitive science* 14.2 (1990): 179-211.

[17] Gildea, D. and Jurafsky, D. "Automatic Labeling of semantic roles". *Computational Linguistics*. 28(3), (2002): 245-288.

[18] Goldberg, Yoav. "A primer on neural network models for natural language processing." Journal of Artificial Intelligence Research 57 (2016): 345-420.

[19] Hoekstra, Heleen, Moortgat, Michael et al. "Syntactic annotation for the spoken Dutch corpus project (CGN)." Language and Computers 37.1 (2001): 73-87.

[20] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[21] Jamtoey, Anne Iren. "Emotion and cognition in political communication". *Paper for the 3rd International Conference on Democracy as Idea and Practice*, Oslo. 2012.

[22] Khan, Aamera ZH, Mohammad Atique, and V. M. Thakare. "Combining lexicon-based and learning-based methods for Twitter sentiment analysis." *International Journal of Electronics, Communication and Soft Computing Science & Engineering* (IJECSCSE) (2015): 89.

[23] Kim, Soo-Min, and Eduard Hovy. "Extracting opinions, opinion holders, and topics expressed in online news media text." *Proceedings of the Workshop on Sentiment and Subjectivity in Text*. Association for Computational Linguistics, 2006.

[24] Koolagudi, Shashidhar G., Nitin Kumar, and K. Sreenivasa Rao. "Speech emotion recognition using segmental level prosodic analysis." *Devices and Communications (ICDeCom)*, 2011 International Conference on. IEEE, 2011.

[25] Lample, Guillaume, et al. "Neural architectures for named entity recognition." *arXiv preprint arXiv*:1603.01360 (2016).

[26] Lisetti, Christine Lætitia, and Fatma Nasoz. "Using noninvasive wearable computers to recognize human emotions from physiological signals." *EURASIP Journal on Advances in Signal Processing* 2004.11 (2004): 929414.

[27] Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.

[28] Liu, H., Lieberman, H., Selker, T.: "A model of textual affect sensing using real-world knowledge." *Technical report, MIT Media Laboratory*, Cambridge, USA. (2003).

[29] Liu, Pengfei, Shafiq R. Joty, and Helen M. Meng. "Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings." EMNLP. 2015.

[30] Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." *arXiv preprint arXiv*:1603.01354 (2016).

[31] Mazzei, Daniele, et al. "Hefes: An hybrid engine for facial expressions synthesis to control human-like androids and avatars." *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference* on. IEEE, 2012.

[32] McSpadden, Kevin. "You now have a shorter attention span than a goldfish." *Time Online Magazine*. Retrieved May 7 (2015): 2016.

[33] Meyer, John-Jules Ch. "Reasoning about emotional agents." *Proceedings of the 16th European conference on artificial intelligence.* IOS Press, 2004.

[34] Mikolov, Tomáš. "Statistical language models based on neural networks." PhD thesis, Brno University of Technology. (2012).

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

[36] van Noord, Gertjan et al. "Lassy Syntactische Annotatie." (2011) http://www.let.rug.nl/vannoord/Lassy/sa-man_lassy.pdf

[37] Ortony, Andrew, Gerald L. Clore, and Mark A. Foss. "The referential structure of the affective lexicon." *Cognitive science* 11.3 (1987): 341-364.

[38] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*-Volume 10. Association for Computational Linguistics, 2002.

[39] Picard, Rosalind W., and Roalind Picard. *Affective computing*. Vol. 252. Cambridge: MIT press, 1997.

[40] Socher, Richard, et al. "Parsing natural scenes and natural language with recursive neural networks." *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.

[41] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631. 2013.

[42] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15.1 (2014): 1929-1958.

[43] Steunebrink, Bastiaan Reinier. *The logical structure of emotions.* Diss. Utrecht University, 2010.

[44] Tang, Duyu, et al. "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification." ACL (1). 2014.

[45] Tripathi, Vaibhav et al. "Emotion Analysis from Text: A Survey"

[46] Den Uyl, M. J., and H. Van Kuilenburg. "The FaceReader: Online facial expression recognition." *Proceedings of measuring behavior.* Vol. 30. 2005.

[47] Wang, Xin, et al. "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory." ACL (1). 2015.

[48] Zhila, Alisa, et al. "Combining Heterogeneous Models for Measuring Relational Similarity." HLT-NAACL. 2013.

[49] Zhou, Jie, and Wei Xu. "End-to-end learning of semantic role labeling using recurrent neural networks." ACL (1). 2015.

[50] Zou, Will Y., et al. "Bilingual Word Embeddings for Phrase-Based Machine Translation." EMNLP. 2013.