Negation Cues Detection Using CRF on Spanish Product Review Texts

Detección de Claves de Negación Usando CRF en El Texto de Revisión de Productos en Español

Henry Loharja[1], **Lluís Padró**[1], and **Jordi Turmo**[1]
[1]Universitat Politècnica de Catalunya
https://www.upc.edu/
{loharja, padro, turmo}@cs.upc.edu

Abstract: This article describes the negation cue detection approach designed and built by UPC's team participating in *NEGES 2018 Workshop on Negation in Spanish*. The approach uses supervised CRFs as the base for training the model with several features engineered to tackle the task of negation cue detection in Spanish. The result is evaluated by the means of precision, recall, and F1 score in order to measure the performance of the approach. The approach was ranked in 1st position in the official testing results with average precision around 91%, average recall around 82%, and average F1 score around 86%.
Keywords: negation cue detection, conditional random field, product review

Resumen: Este artículo describe el enfoque de detección de claves de negación diseñado y construido por el equipo de la UPC que participa en textit Taller NEGES 2018: Identificación de Claves de Negación. El enfoque usa el CRF supervisado como la base para el entrenamiento del modelo con varias características diseñadas para resolver la tarea de detección de claves de negación en español. El resultado se evalúa mediante el método de precisión, exhaustividad y Valor-F para medir el rendimiento del enfoque. El enfoque se clasificó en primero posición en los resultados de las pruebas oficiales con una media de precisión cerca del 91 %, una media de exhaustividad cerca del 82 % y una media de Valor-F cerca del 86 %.
Palabras clave: detección de clave de negación, campo aleatorio condicional, revisión del producto

1 *Introduction*

This paper describes the negation cue detection model approaches presented by UPC's team for the NEGES 2018 workshop task 2 (negation cues detection) (Jiménez-Zafra et al., 2018a). The aim of the task is to automatically detect negation cues in product review texts in Spanish. To do this, the participants must develop a system able to identify all the negation cues present in the documents. The SFU ReviewSP-NEG corpus (Jiménez-Zafra et al., 2018b) will be used to train and test the systems. The approach we develop relies on a supervised learned model using Conditional Random Fields (written as CRF in the following contents) as the core with specially engineered features for the detection of negation cues in Spanish. The approach is then implemented in Python and we use NLTK[1] as the toolkit to build the system. The result is measured using the widely used performance measurement of precision, recall, and F1 score.

The article is organized as follows. Section 2 describes the approach used to learn the negation cues detection model. Section 3 describes the system built based on the approach explained in the previous section and the details of the implementation. The results achieved by our approach are presented and briefly analyzed in Section 4. Finally, Section 5 gives conclusion about the work that has been done.

[1]NLTK – the Natural Language Toolkit (Bird, Klein, and Loper, 2009)

Proceedings of NEGES 2018: Workshop on Negation in Spanish, pages 49-54
Seville, Spain, September, 18, 2018

2 Negation Cues Detection Approach

Before describing the approach, let us begin by addressing some definitions. A *negative sentence* n is defined as a vector of words $(w_1, w_2, ..., w_n)$ containing one or more negation **cues**, where the latter can be a word (e.g. *no*), a morpheme (e.g.*in*-capaz) or a multi-word expression (e.g. *ya no, todavía no*) which inherently expressing negation. The goal of negation cue detection is to predict vector c given the sentence n where $c \in \{1,0\}^{|n|}$ is a vector of length same with the length of n so that $c_i = 1$ if w_i is part of the negation cue and $c_i = 0$ otherwise.

It is possible that more than one negation cue can appear inside a sentence. In Spanish, one of the special characteristic of negation cue is that a cue can consist of more than one word, not necessarily consecutive. This special characteristic increases the complexity of detecting whether two words recognized as cue are indeed two separated cues or are actually the same non-contiguous cue. This also makes negation cue detection in Spanish a more challenging task compared to negation cue detection in English because that case is scarce in English.

The approach we use for this work will be one of the state of the art approach: a CRF based negation detection. We try to reproduce the approach used by previous works (Agarwal and Yu, 2010) which are using CRF as its base and we use the corpus given by the task in order to see how the approach perform with the data provided in Spanish. Conditional random fields (CRFs) are a type of discriminative undirected probabilistic graphical model used for structured prediction (Lafferty, McCallum, and Pereira, 2001). The most important feature of a CRF model is that it can take context into account: the linear chain CRF predicts sequences of labels for sequences of input samples. Thus, the model does not work with local probabilities like $p(y_t|x_t)$ where t is the position of x within the sequence, instead, it estimates the conditional probability of the whole sequence:

$$p(y|x) = \frac{1}{Z(x)} exp\{\sum_{t} \sum_{j=1}^{K} \lambda_j f_j(x, y_t, y_{t-1})\}$$

The estimation of weights (λ_j) for each feature f_j is carried out by maximizing the conditional log likelihood:

$$\max_{\lambda} l(\lambda) = max \sum_{i=1}^{N} p(y^{(i)}|x^{(i)})$$

.

where N is the number of observation sequences $x^{(i)}$ and label sequences $y^{(i)}$.

Training CRFs might be time-consuming for some tasks since the time needed for training depends quadratically on the number of class labels and linearly on the number of training instances and the average sequence length. However, state-of-the-art solutions use CRF models for many NLP tasks where time consumption is still tolerable.

As discussed before, the goal of negation cue detection is to obtain vectors which represent the sentence and give information whether the token or words which are part of the sentence is a part of the negation cue in a value. Using the knowledge of named entity recognition, we can infer that negation detection is a type of NER in which we would like to recognize entities that are parts of negation. In other words, we would like to classify whether each words inside a sentence is part of negation cue or not a part of it. From this, we define a three-class classification problem for each word which we observe: Begin-Cue(B-C), Inside-Cue(I-C), or Out(O). A word classified as Out is not part of a cue. In order to handle cues which consist of more than one word, we give two kind of classification for the cues which are Begin-Cue for the first words that identify that start of a cue and Inside-Cue for the rest of the words of a cue which are not the first word but is still identified as part of the same cue.

3 Negation Cues Detection System

3.1 Data Preprocessing

For the preliminary, we do some preprocessing to the data in the corpus provided in order to match the input format of the system we built. The corpus provided by this task is using CoNLL format. Each line corresponds to a token or word and each annotation is provided in a column with empty lines indicate end of sentence. We produce two set of data with different format with respect to their usage for each step:

1. Data format with BIO tagging (Ramshaw and Marcus, 1999) in

order to be used as input for training. The annotated token is tagged with "B-C" if it is in the beginning of negation cue; tagged with "I-C" if it is part of the negation cue but not the first word of the cue; and tagged with "O" if it is outside of the cue. One of the examples of sentence in this format is:

- `El|O coche|O funciona|O estupendamente|O ,|O es|O muy|O manejable|O ,|O por|O cierto|O ,|O casi|B-C no|I-C consume|O gasolina|O algo|O que|O para|O mi|O es|O muy|O importante|O .|O`

2. Raw data format without any tagging in order to be used for testing input.

- `Las ruedas a los 15000 kms las tuve que cambiar , todas , las cuatro , por ser de una marca coreana , que no da mucho resultado .`

After the preprocessing is done, the documents is ready to be used as input for the next respective steps. This preprocessing part did not alter any important information contained in the data as the purpose is only to change the format in order to make it easier to be used in the following steps.

3.2 Baseline System

Before implementing the approach we have explained before in the system, we developed a baseline system to be used as starting point and a comparison. Our aim is to see whether the approach we have will perform better than a baseline approach which used simple techniques. To reach this, we use the baseline system as comparison with the system we develop using the approach we propose. The baseline system we developed uses simple techniques which are common such as dictionary lookup combined with some rules for detecting negation cues.

The first thing we did in this baseline was to create a dictionary based on the training dataset from the corpus. We collected all the words which are tagged as negation cues from all the documents in the training dataset together with their frequency. After having sorted the collected negation cues based on the frequency from the most to the least, we chose the top 25 words with the most frequency. These 25 words became the dictionary of negation cues in our baseline system. We also developed several rules to capture the characteristics of negation cues in Spanish which we have explained before. These rules are used to decide whether more than one cues which appear in a sentence is actually part of a cue or separated cues. The rule checked whether a word is in a list of special word which we created and then check if it fulfill condition of having another cues that precede it. Here are the algorithm from baseline system which describe the rule:

```
if word in DICTIONARY then
    if word in SPECIAL then
        if exist cue before then
            word is part of cue
        end if
    else
        word is new cue
    end if
end if
```

After having implemented the combination of dictionary lookup and rules we developed, we then use the baseline system to tag the documents from development testing dataset. We use the result as the preliminary result to be later compared with the result from the system we developed using our proposed approach. By doing this, we could see whether the approach we have can give more advantage compared to using simple techniques.

3.3 Learning The Model for Negation Cue Detection using CRF

The system we built use a toolkit named NLTK which is a Python based toolkit for building Python programs to work with human language data. NLTK provides easy-to-use text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. One of the modules in NLTK is an CRF tagger which can be used for the tagging of text using *Python CRFSuite*[2] as it's core. This module are what we mainly used in our approach for negation detection by adapting a point of view

[2]Python CRFSuite -Python bindings to CRFSuite (Okazaki, 2007)

of named entity recognition. There are two

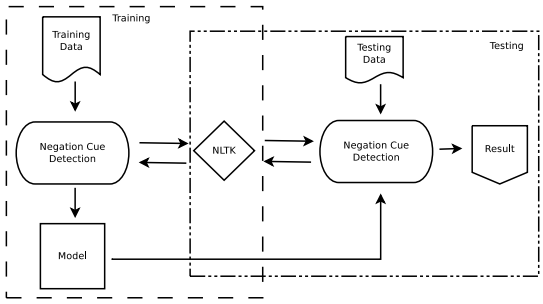

Figure 1: Flow that describe negation detection approach in the system.

main parts in the system we built: Training and testing. Training is the part in which we use the CRF tagger module in NLTK to train a model by using the training data which we have prepared before. The result of the training part is a model for detecting negation. The training process will use orthographic feature set which is designed for negation cue detection and to capture the characteristics of negation cue in Spanish. The simplest and most clear feature set is the vocabulary from the training data. We also include the information about part of speech as feature in order to enrich the feature set. Generalizations over how the words written (capitalization, affixes, etc.) are also important information that are included as features. The present approach includes training vocabulary, several orthographic features based on regular expressions as well as prefixes and suffixes in the character length ranged from two to four. To model localization context, neighboring words in the window [-6,1] are also added as features. This size of window is selected from several experiment using various window size to acquire optimum result. We use bigram in the process of including the information about localization of six word before and one word after the word being observed. Here are the complete set of features used in the training:

1. WORD: the vocabulary of word.
2. POS: the information of part of speech of the word.
3. INIT_CAP: word starts with capitalization.
4. ALPHANUM: word consists of alphanumeric characters.
5. HAS_NUM: word contains number.
6. HAS_CAP: word contains capitalized letter.
7. HAS_DASH: word contains dash (-).
8. HAS_US: word contains underscore (_).
9. PUNCTUATION: word contains punctuation.
10. SUFn: suffixes in the n character length ranged from two to four.
11. PREFn: prefixes in the n character length ranged from two to four.
12. 2GRAMBEFORE: bigram of up to 6 word before the observed word.
13. 2GRAMAFTER: bigram of up to 1 word after the observed word.
14. BEFOREPOS: the information of part of speech of up to 6 word before the observed word.
15. AFTERPOS: the information of part of speech of up to 1 word after the observed word.
16. SPECIAL: word is one of the special words in the special dictionary. The words we included as special words are: "nada", "ni", "nunca", "ningun", "ninguno", "ninguna", "alguna", "apenas", "para_nada", and "ni_siquiera". These words have more tendency to be part of negation cue with multiple words. This feature is included in order to capture the characteristic of negation cue that can consist of more than one words which are separated by other non-cue words in between.

By using the features mentioned above, we do the training using the given data and CRF module in NLTK to produce the model which can be used to detect the negation cue in Spanish. The parameters for training the CRF are the default parameters used in NLTK toolkit. This model will be used as one of the input for the next step which is testing. Testing is the process of detecting negation from the testing data (data in which negations are not annotated or raw data) by using the model which we get from the training process as the knowledge base. The result of the testing process is an annotated version of testing data in which words in each sentence are classified into either part cue or outside of them. The result we obtain after

the testing process will be in the format of BIO tagged since this is the format which we use to represent our data. Related to that, we do some post-processing to change the format of the result into the same original format as the input (training data). We use the original data format of CoNLL and then add the information of negation cue which we obtain from the testing.

4 Results

After finished with the testing process, we will obtain the result of the negation cue detection as annotated documents of testing data we provide as input. In order to evaluate the performance of the approach used in the system, we will use recall, precision, and f1 score measurement. We use the evaluation script provided by the organizers to make sure that our output match the requirement. In the first phase, we use the development testing data which is provided in order to measure the performance of our system. We perform the testing on each document in the development testing dataset which are divided based on the domain. Each document is processed separately and also evaluated separately. To give a general view of the performance, we also calculate the micro average of the whole result from the development testing. Table 1 shows the result of baseline system we have obtained using the development testing data meanwhile Table 2 shows the result of system based on our proposed approach using the same development dataset.

Domain	**Precision**	**Recall**	**F1**
Coches	88.89	85.11	86.96
Hoteles	86	70.49	77.48
Lavadoras	94.74	80	86.75
Moviles	94.9	85.32	89.86
Musica	79.31	88.46	83.64
Ordenadores	85.71	69.23	76.59
Libros	88.65	86.81	87.72
Peliculas	92.55	79.09	85.29
Micro Average	90.06	81.31	85.32

Table 1: Measurement result of development testing using baseline system

As can be observed from Table 1 and 2, the result using our proposed approach gives better result compared to the baseline system which use simple techniques. The result also gives a fairly high value of performance with most of them reach over 80%. Especially in precision, the average reach more than 90%. This is possible due to a fairly simple task of detecting negation cue detection. Most of the cues consist of word such as "no","ni","nada" and several other words which describe negation with little variability of vocabulary. This leads to a fairly easy detection of cues and the small number of false positives. On the other hand, the recall have much lower result with some reach even lower than 80%. This happens due to the higher number of false negatives caused by the difficulty of detecting non-contiguous multi-token cues. In most of the cases of false negative, our system has difficulties to detect such cases, for example:

Domain	**Precision**	**Recall**	**F1**
Coches	83.33	74.47	78.65
Hoteles	96.08	80.33	87.5
Lavadoras	97.3	80	87.81
Moviles	95.1	88.99	91.94
Musica	83.33	96.15	89.28
Ordenadores	89.13	78.85	83.68
Libros	90.85	89.58	90.21
Peliculas	92.93	83.64	88.04
Micro Average	91.97	84.85	88.14

Table 2: Measurement result of development testing using CRF based approach

- **No** es cosa del paralelo **ni** del equilibrado.

In the example, *no...ni* is a negation cue meanwhile our system recognize them as two separated cues. Another kind of false negative is the opposite, where two separated cues is recognized as one cue. Those cases contribute to most of the false negatives in the development testing result.

The official testing result measurement can be observed in Table 3. This result is obtained using the model we have and the official testing dataset provided by the organizers. The evaluation is done directly by the organizers and we receive the measurement result as can be seen in Table 3 after we submit our testing result.

Based on the evaluation from organizers, our result is ranked first compared to other

Domain	Precision	Recall	F1
Coches	95.08	85.29	89.92
Hoteles	94	79.66	86.24
Lavadoras	94.74	78.26	85.72
Moviles	89.8	77.19	83.02
Musica	92.96	75.86	83.54
Ordenadores	91.36	91.36	91.36
Libros	84.19	84.52	84.35
Peliculas	89.68	85.28	87.42
Average	91.48	82.18	86.45

Table 3: Measurement result of official testing

participants in the same task. As can be seen on the table, the official testing result follows the same pattern as the development testing result with higher precision and lower recall. Even though we can't observe the cases happening in official testing result, we can infer that similar cases in development testing probably also can be found by looking at the result. The percentage also have almost similar value with precision reach around 91%, recall around 82%, and F1 score around 86%. The average of result in official testing has slightly lower value compared to the one in development testing but the difference is not significant.

5 Conclusion

In this article we have described the approach and system we built for the participation in NEGES 2018: Workshop on Negation in Spanish task 2 of negation cues detection for Spanish product review texts. Our approach to detect the negation cues consisted of a supervised approach combining CRF and several features for negation cue detection in Spanish for training the model. The model will then be used to classify whether a word in the observed data or testing data is a part of negation cue or not. This approach was ranked in 1st position in the official testing results with average precision around 91%, average recall around 82%, and average F1 score around 86%.

Acknowledgements

This works has been partially funded by the Spanish Goverment and by the European Union through GRAPHMED project (TIN2016-77820-C3-3-R and AEI/FEDER,UE.)

References

Agarwal, S. and H. Yu. 2010. Biomedical negation scope detection with conditional random fields. 17:696–701, 11.

Bird, S., E. Klein, and E. Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Jiménez-Zafra, S. M., N. P. Cruz-Díaz, R. Morante, and M. T. Martín-Valdivia. 2018a. Resumen de la Tarea 2 del Taller NEGES 2018: Detección de Claves de Negación. In *Proceedings of NEGES 2018: Workshop on Negation in Spanish*, volume 2174, pages 35–41.

Jiménez-Zafra, S. M., M. Taulé, M. T. Martín-Valdivia, L. A. Ureña-López, and M. A. Martí. 2018b. SFU Review SPNEG: a Spanish corpus annotated with negation for sentiment analysis. A typology of negation patterns. *Language Resources and Evaluation*, 52(2):533–569.

Lafferty, J., A. McCallum, and F. C. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Okazaki, N. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

Ramshaw, L. A. and M. P. Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, pages 157–176.