2010 Second International Conference on Knowledge and Systems Engineering

Sentiment Analysis for Vietnamese

Binh Thanh Kieu
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University Hanoi
E-mail: binhkt.vnu@gmail.com

Son Bao Pham
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University Hanoi,
Information Technology Institute
Vietnam National University Hanoi
E-mail: sonpb@vnu.edu.vn

Abstract — **Sentiment analysis is one of the most important tasks in Natural Language Processing. Research in sentiment analysis for Vietnamese is relatively new and most of current work only focus in document level. In this paper, we address this problem at the sentence level and build a rule-based system using the Gate framework. Experimental results on a corpus of computer products reviews are very promising. To the best of our knowledge, this is the first work that analyzes sentiment at sentence level in Vietnamese.**

Keywords - Sentiment Analysis, Opinion Mining, Text Mining

I. INTRODUCTION

In recent years, along with the rapid growth of the Internet, textual information on the web is becoming larger and larger. Generally, textual information is often classified into two main types: facts and opinions. Most current information processing techniques (search engines) works with facts. Facts can be expressed with topic keywords. However, search engines do not search for opinions. An example for this kind of information is the product reviews. This information can be collected from manufacturers or users. Manufacturers use opinions for building business strategy. A sentiment analysis system about product's quality is expected to meet the need of both the users and the manufacturers.

Technically, each sentiment analysis system can often be divided into two parts: identifying words and phrases that hold opinions and classifying sentence or document according to the opinions. Unlike the classification by types or subject, the classification by sentiment requires the understanding of the emotional trend in the article. Some challenging aspects in sentiment analysis include the identification of opinion terms, the intensities of sentiment, the complexity of sentences, words in different contexts and sentiment classification for the complex articles etc.

In this paper, we propose a rule-based method for constructing automatic evaluation of users' opinion at sentence level. Using a rule-based approach is a natural choice since there is no publicly available corpus for Vietnamese sentiment analysis. Our system is built on GATE [1] - a framework for developing components of natural language processing. Our system focuses on the domain of computer products (laptop & desktop).

We will present related work on sentiment analysis in section 2 and describe our system in section 3. Section 4 will show some experimental results and error analysis. Finally, section 5 will give concluding remarks and pointers to future work.

II. RELATED WORK

For the last decade, sentiment mining has become a hot subject among natural language processing (NLP) and information retrieval (IR) researchers [9]. Though the works on sentiment mining all have different focuses, emphasizes and objectives; nevertheless, they generally consists of the following three steps: sentiment words or phrases identification, sentiment orientation identification and sentiment sentence or document classification.

Sentiment words or phrases identification focuses on content words (nouns, verbs, adjectives and adverbs) where most of the works use part-of-speech (POS) to extract them [4][8][11][16]. Other natural language processing techniques such as stop words removal, stemming and fuzzy matching are also used in the preprocessing stage to extract sentiment words and phrases.

In the work about sentiment orientation identification, there are many approaches proposed. Hu and Liu [8] applied POS tagging and some natural language processing techniques to extract the adjectives as sentiment words. Experimental result of their opinion sentence extraction has a precision of 64.2% and a recall of 69.3%. Fellbaum [5] uses WordNet to determine whether the extracted adjective has a positive or negative polarity. The pointwise mutual information (PMI) is used by Church and Hanks [2] and Turney [15] to measure the strength of semantic association between two words. Nasukawa and Yi [11] also consider verbs as sentiment expressions for their sentiment analysis. They use HMM-based POS tagger [10] and rule-based shallow parsing [12] for preprocessing. They then analyze the syntactic dependencies among the phrases and look for phrases with a sentiment term that modifies or is modified by a subject term.

The task of sentence or document sentiment classification is to classify a sentence or document according to its polarity into different sentiment categories – positive or negative with neutral category added sometimes. Hu and Liu [8] predict the orientation of the opinion sentence in their study of customer reviews. Turney [16] used a simple unsupervised algorithm to classify reviews in different domains as recommended or not recommended and then do sentiment words (phrases) extraction based on Hatzivassiloglou and McKeown's [7] approach and orientation identification based on Turney's

978-0-7695-4213-3/10 $25.00 © 2010 IEEE
DOI 10.1109/KSE.2010.33

[15] approach. The averaged classification accuracy of the reviews in different domains is 74.39%. Pang [13] used supervised machine learning to classify movie reviews. Without classifying individual sentiment words or phrases, they extract different features from the review and use Naive Bayes, Maximum Entropy and Support Vector Machine to classify the reviews. They achieved accuracies between 78.7% and 82.9%.

III. Our System of Analyzing Users' Opinions

Most of approaches in sentiment analysis are language and domain dependant. Our approach analyzes product's features sentiment and classifies it into two categories: positive or negative. In the process of data collection, we realize almost all sites were discussing only one product in each thread, so we assume that only one product is the target of review in a document. However there are many discussions about different features of the product in one document.

A. Data and annotation

This is the first step to build our rule-based system. One constraint is that most Vietnamese product reviews available online are about electronic devices. In addition, the product feedbacks and reviews are often written by teens that use special language including new terms, abbreviation, mixed with foreign terms etc. Our data is mainly taken from an online product-advertising page [17] with computer category (laptops & desktop). In the future we will extend the data to include other products such as mobile phones and automobiles. After we collected the data, we preprocess the data such as: standardizing short words ("wa", "ko").

The corpus we have collected contains about 3971 sentences in 20 documents corresponding to 20 products. With the collected corpus, we use Callisto[1] annotation tool [3] to mark up annotations at different levels to do our sentential sentiment analysis. We use this process to obtain an annotated corpus and also to incrementally create the rules. At the word level, we have two annotations PosWord (positive word) and NegWord (negative word). For sentence level, we use PosSen (positive sentence), NegSen (negative sentence) and MixSen (mixed sentence) annotations to distinguish sentences with positive, negative and both positive and negative sentiment respectively. To handle sentences that have implicit sentiment via comparing different products, we use CompWord (comparison word) and CompSen (comparison sentences) annotations.

B. System Overview

Our systems are built based on three main components: sentiment words or phrases identification, sentiment orientation identification and sentential sentiment classification. These three components are executed in the following order:

1. *Preprocessing:* Word segmentation and POS tagger.
2. *Word processing:* Identify words, phrases and sentiment words and phrases.
3. *Sentence processing:* Classify sentential sentiment.
4. *Evaluate product features* based on the classified sentences.

Let's look at the following input sentence:

"HP dv 4 có thiết kế bắt mắt, ưa nhìn tuy nhiên giá quá cao."

HP dv 4 has an eye-catching, nice design but is too expensive.

In the preprocessing step, we use word segmentation and POS tagger:

"<X>HP dv 4</X> <Vts>có</Vts> <Vt>thiết kế</Vt> <V>bắt mắt</V>, <A>ưa nhìn</A> <Cc>tuy nhiên</Cc> <Na>giá</Na> <Jd>quá</Jd> <An>cao</An>."

After preprocessing, we identified sentiment words and phrases:

"HP dv 4 có <kieudang>thiết kế</kieudang> <PosWord>bắt mắt</PosWord>, <PosWord>ưa nhìn</PosWord> tuy nhiên <gia>giá</gia> quá <NegWord>cao</NegWord>."

We divided sentences into simple sentence (or clauses) and classified simple sentences' sentiment:

"<PosSen>HP dv 4 có thiết kế bắt mắt, ưa nhìn</PosSen> tuy nhiên <NegSen>giá quá cao.</NegSen>"

Finally, we summarized overall products features' sentiment:

Kiểu dáng (design): 1/0 (#positive/#negative)
Giá (cost): 0/1 (#positive/#negative)

The effectiveness of the GATE framework for NLP tasks has been proven through many researches, so we decided to build our Vietnamese sentiment analysis system as plugins in GATE. The architecture of the system is shown in Figure 1 with the following three components:

1. *Preprocessing:* Vietnamese word segmentation and POS tagger.
2. *Dictionaries:* matching words in the positive word dictionary, negative word dictionary etc.
3. *Rules:* word identification, sentence classification, and features evaluation.

C. Preprocessing

A distinctive feature of the Vietnamese language is word segmentation. An English word is identified by space characters, but words in Vietnamese are different. A word in Vietnamese language can consist of more than one monosyllable. For example the following sentence:

"Học sinh học sinh học."

may be word-segmented as follows:

"Học_sinh học sinh_học." (Students study biology) or
"Học sinh học sinh học." (Study biology biology)

In our system, we reuse an existing Coltech.NLP.tokenizer plugin [14] for word segmentation and POS tagging.

D. Dictionaries

During the process of annotating the corpus using Callisto, we created a number of dictionaries, which can be divided into two groups:

[1] [1] http://callisto.mitre.org/download.html

1. Dictionaries containing names related to features recognition:

a. Dictionary of words related to configuration features of computer products such as: *cấu hình* (configuration), *hệ thống* (system), *vi xử lý* (CPU) etc.

b. Dictionary of words related to "*kiểu dáng*" (appearance) feature: *kiểu dáng* (appearance), *thiết kế* (design), *thân hình* (body), *kích thước* (size), *màu sắc* (color) etc.

2. Dictionaries containing words used to develop rules to identify features' sentiment:

a. Positive word dictionary: tốt (good), tuyệt vời (excellent), hoàn hảo (perfect), hài lòng (satisfying) etc.

b. Negative word dictionary: xấu (ugly), đắt (expensive), thô (rough), phàn nàn (complain), thất vọng (disappointing) etc.

c. Reverse opinion word dictionary: không thể (cannot), không quá (not too) etc.

E. Rules

There are four types of rules:

1. Dictionaries lookup words correction.
2. Sentiment word recognition.
3. Sentential sentiment classification
4. Features evaluation

We use Gate's Jape grammar to specify our rules. A Jape grammar allows one to specify regular expression patterns over semantic annotations. Bellows is an example of a JAPE rule to recognize one type of positive words:

Rule: rulePositive1

Priority: 1

(

(StrongWord)

({Word.category=="O"})?

({Lookup.majorType=="positive"}) :name

)

-->:name.PosWordFirst = {kind = "StrongWord + <O>? +<PosWord>", type="Positive", rule = "Positive recognition"}

In the first step, we remove monosyllables appearing in dictionaries but are not words and do not carry the correct meaning in context. For example:

"Macbook Pro MB471ZPA có **giá** quá cao. Tuy nhiên chiếc Laptop này vẫn được đánh **giá** cao."

"Macbook Pro MB471ZPA has a too high **price**. However, this Laptop is still strongly **recommended**."

Because our dictionaries include the word "giá" to refer to the feature "giá" (price) of products so it would be incorrect to identify "giá" in the word ″đánh giá″ (recommend) as a feature "giá". This could simply be fixed by overwriting the result of word segmentation over dictionaries lookup.

In sentiment word recognition step (an example in Figure 2), sentiment words are determined based on dictionaries but there are many cases where simply matching dictionaries without considering the context gives a wrong result. For example "thời trang" (fashion) is a sentiment word in the sentence "Phong cách rất **thời trang**" (very **fashionable** style) but not a sentiment word in the sentence "Thiết kế của máy có nét **thời trang** giống với chiếc xe ô tô" (The **fashion** feature of this laptop is similar to that of a car). There are also cases where a word can bring both positive and negative sentiment depending on context. For example, the word "cao" (high) is positive if it talks about computer configuration but is negative when talking about price.

Contextually, it is easy to notice that sentiment words usually appear after some adverbs. For example, positive sentiment words (PosWord) go with "rất" (very), "siêu", "khá", "cực", "đáp ứng" while negative sentiment words (NegWord) go with "dễ", "hơi", "gây", "bị". We use the following pattern to recognize sentiment words:

<StrongWord> + <Adv> + <word in sentiment dictionaries> -> opinion word

When user uses multiple sentiment words for describing a features such as in the following example:

"Laptop cho doanh nhân Acer Aspire 3935 sử dụng thiết kế phá cách, hiện đại."

"Acer Aspire 3935 laptops for business use an innovative and modern design"

We use the following pattern:

<Opinion word> (<conjunction: , và (and) hay (or) …> <Opinion word>)*

Another important scenario is when users use words that reverse the sentiment of the following statement. We simply use the following rule to handle this case:

<Reverse Opinion> < positive word (negative word)> -> < negative word (positive word)>

In addition, we also create other rules based on POS tags using unit testing to ensure consistency between new rules and the data already correctly identified by existing rules.

The sentiment sentence classification step consists of two main subtasks:

- Simple sentence (or clauses) split
- Sentiment sentence classification: PosSen (positive sentence), NegSen (negative sentence), MixSen (mixed sentence) and CompSen (comparison sentence).

Compound sentences may contain more than one clause discussing several features of a product. The *simple sentence split* step is to identify compound sentences and split them into separate simple sentences. We create rules to determine simple sentences using connective words. After this step, all sentences are considered simple and talk about only one feature per sentence.

For sentence classification, there are 4 main types: positive sentence, negative sentence, mixed sentence and comparison sentence [6]. Positive sentences (PosSen) are assumed to include only positive words (PosWord). Negative sentences (NegSen) are assumed to include only negative words (NegWord). And mixed sentences

(MixSen) contain both positive and negative sentiment words. Among sentences not containing any sentiment words, we identify sentences containing comparison expressions and label them as CompSen. With comparison sentences, because the sentences often compare one product with another product, we assume the target product of the document is always mentioned first and the nature of the comparison corresponds to the sentiment. In particular, if it is a better or worse comparison then it is of positive or negative sentiment respectively. In effect, CompSen sentences will be converted to PosSen and NegSen where appropriate.

Overall features evaluation is based on the result of simple sentence classification. For positive and negative sentences, it is quite straightforward as we only have to identify the feature mentioned in the sentence and deem the sentiment of sentence to be the sentiment of the feature. For mixed sentences, we use an assumption that they normally have the following format *<Feature>* *<Opinion>* *<Feature>* *<Opinion>*. Therefore we associate each sentiment with the nearest preceding feature.

Feature evaluation simply counts how many positive and negative sentences containing the feature and output the ratio between the number of positive and negative sentences. This ratio captures how users think about the feature.

IV. EXPERIMENTS

We collected a corpus of computer products reviews and feedbacks and manually annotated all the data using the annotations described in section 3.1. The corpus consists of 3971 sentences in 20 documents corresponding to 20 products. We divided the corpus into 2 parts: the training set and test set. The training set contains 16 documents (3182 sentences), which is used to create dictionaries and rules for identifying all the annotations. The test set contains 4 documents and it is used to test the performance of our rule-based system.

We run the experiments at three levels: word, sentence and features. For word and sentence level evaluation, we just compare the annotation at corresponding levels posted by the system with the manually created annotation in the test data.

A. Experiment for sentiment word recognition

At the word level, we evaluate how well the system can identify PosWord and NegWord from the test data using the standard Precision, Recall and F-measure measures. Table 1 and Table 2 show the results of the system running on training data and test data respectively. It appears that the rule-based system generalizes quite well for sentiment word recognition task, as the F-measure on the test data is comparable to training data.

Table 1 – Result of sentiment word recognition on training data

	#Annotation	#System Annotation	#True annotation	Precision	Recall	F-measure
PosWord	441	376	334	88.83 %	75.74 %	82.28 %
NegWord	153	122	93	76.23 %	60.78 %	68.51 %
All	598	502	431	85.86 %	72.07 %	78.97 %

Table 2 - Result of sentiment word recognition on test data

	#Annotation	#System Annotation	#True annotation	Precision	Recall	F-measure
PosWord	300	237	214	90.30 %	71.33 %	79.70 %
NegWord	60	62	42	67.74 %	70.00 %	68.85 %
All	362	301	258	85.71 %	71.27 %	77.83 %

B. Experiment for sentential sentiment classification

At the sentence level, we evaluate the system on the task of labeling PosSen, NegSen and MixSen annotations. Table 3 and Table 4 show the F-measures of the system for recognizing these three annotations on training and test data respectively.

Table 3 - Result of sentential sentiment classification on training data

	#Annotation	#Annotation	#True annotation	Precision	Recall	F-measure
PosSen	231	218	154	70.64 %	66.67 %	68.60 %
NegSen	97	96	67	69.79 %	69.07 %	69.43 %
MixSen	9	26	7	26.92 %	77.78 %	40.00 %
All	340	343	231	67.35 %	67.94 %	67.64 %

Table 4 - Result of sentential sentiment classification on test data

	#Annotation	#System Annotation	#True annotation	Precision	Recall	F-measure
PosSen	157	157	99	63.06 %	63.06 %	63.06 %
NegSen	49	45	34	75.56 %	69.39 %	72.34 %
MixSen	5	21	3	14.29 %	60.00 %	23.08 %
All	212	224	137	61.16 %	64.62 %	62.84 %

It can be seen that the performance for identifying sentential sentiment is not very high compared to sentiment words. It is partly due to the simple heuristic we use to identify sentential sentiment based solely on sentiment words. The MixSen also proves to be much more difficult to recognize compared to PosSen and NegSen.

C. Features Evaluation

For every product, we evaluate the performance of the system on each feature of the product. In this experiment, we are going to evaluate five features: "vận hành" (operation), "cấu hình" (configuration), "màn hình" (monitor), "giá" (price), and "kiểu dáng" (appearance). The output of the system for each feature is the ration *a/b* where *a* and *b* are the number of positive and negative sentences mentioning the feature respectively. For example 15/10 means 15 positive sentences discuss the feature and 10 negative sentences talk about the feature.

We define the following measure for a feature:

Degree of positive sentiment = (number of PosSen) / (number of PosSen + number of NegSen)

Deviation = | System's degree of positive sentiment – correct degree of positive sentiment |

*Correctness = (1 - Deviation)*100%*

The correctness for a product is the averaged value of the correctness measure of the product's features.

Table 5 and Table 6 show the correctness of system when analyzing sentiments for some products on training data and test data respectively.

Table 5 – Result of features evaluation on training data

Product	Correctness
Acer Aspire 3935	92.83%
Apple Macbook Air MB543ZPA	84.26%
Acer Aspire AS4736	96.11%
All	91.07%

Table 6 - Result of features evaluation on test data

Product	Correctness
Dell Inspiron 1210	84.32 %
Compaq Presario CQ40	89.99%
HP Pavilion dv3	92.11%
All	88.81%

Even though the system's performance on sentence level is not very high, but looking at the product as a whole it is quite reasonable with the averaged correctness of nearly 90%.

V. CONCLUSION

We have built a rule-based sentiment analysis system for Vietnamese computer product reviews at sentence level. Our system looks at features of a product and output the ratio of the number of positive and negative sentiments towards every feature. To the best of our knowledge, this is the pioneering work for Vietnamese sentiment analysis at sentential level.

Even though the system achieves F-measures of around 77% and 63% for word and sentence levels respectively, the overall result for a product is of 89% correctness. While the measure used for evaluating performance of the system on the product level is subjective, it is indicative of the effectiveness and potential of our system.

In the future, we plan to collect a larger data set with more diverse domains and combine our system with machine learning approaches.

ACKNOWLEDGEMENT

This work is partly supported by the research project No. QG.10.39 granted by Vietnam National University, Hanoi and the IBM Faculty Award 2009 for the second author.

REFERENCES

[1] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. 2002. "GATE, A Framework and Graphical Development Environment for Robust NLP Tools and Applications". Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.

[2] K. W. Church, P. Hanks. 1989. "Word association norms, mutual information and lexicography". Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics.1989, Vancouver, B.C., Canada, pp76–83.

[3] D. Day, C. McHenry, R. Kozierok, L. Riek. 2004. "Callisto: A Configurable Annotation Workbench". In Proceedings of the Fourth International Conference on Language Resources and Evaluation. (LREC 2004). ELRA. May, 2004.

[4] X. Ding, B. Liu, L. Zhang. 2009. "Entity Discovery and Assignment for Opinion Mining Applications". Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

[5] C. Fellbaum. 1998. "WordNet: an electronic lexical database". MIT Press.

[6] M. Ganapathibhotla and B. Liu. 2008. "Mining Opinions in Comparative Sentences". Proceedings of the 22nd International Conference on Computational Linguistics.

[7] V. Hatzivassiloglou and Kathleen R. McKeown. 1997. "Predicting the Semantic Orientation of Adjectives". Proceedings of the 8th conference on European chapter of the Association for Computational Linguis- tics. 1997, Madrid, Spain.

[8] M. Hu and B. Liu. 2004. "Mining and summarizing customer reviews". Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining. Aug. 22–25, 2004, Seattle, WA, USA.

[9] A. Kao and Stephen R. Poteet. "Natural Language Processing and text mining". April 2006. Chapter 2.

[10] C. Manning and H. Schutze. 1999. "Foundations of Statistical Natural Language Processing". MIT Press, Cambridge, MA.

[11] T. Nasukawa and J. Yi. 2003. "Sentiment Analysis: Capturing Favorability Using Natural Language Processing". Proceedings of the 2nd international conference on Knowledge Capture.

[12] Mary S. Neff, Roy J. Byrd, and Branimir K. Boguraev. 2003. "The Talent System: TEXTRACT Architecture and Data Model". Proceedings of the HLT-NAACL2003 Workshop on Software Engineering and Architecture of Language .

[13] B. Pang, L. Lee and S. Vaithyanathan. 2002. "Thumbs up? Sentiment classification using machine learning techniques". Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP-02).

[14] D. Duc Pham, G. Binh Tran, Son Bao Pham. 2009. "A Hybrid Approach to Vietnamese Word Segmentation using Part of Speech tags". International Conference on Knowledge and Systems Engineering.

[15] P. Turney. 2001. "Mining the Web for synonyms: PMI-IR versus LSA on TOEFL". Proceedings of the 12th European Conference on Machine Learning. Berlin: Spinger-Verlag, pp. 491–502.

[16] P. Turney. 2002. "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews".

Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02). Jun. 2002, Philadelphia, PN, USA, pp.417–424.

[17] http://tinvadung.vn

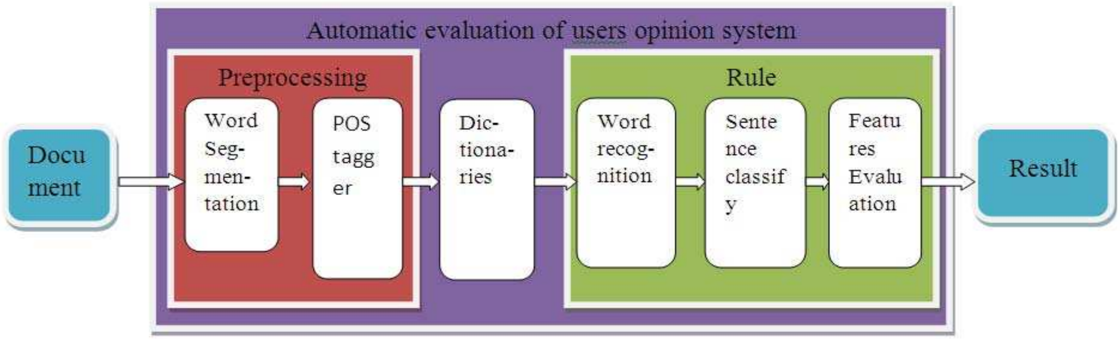

Figure 1 – System overview

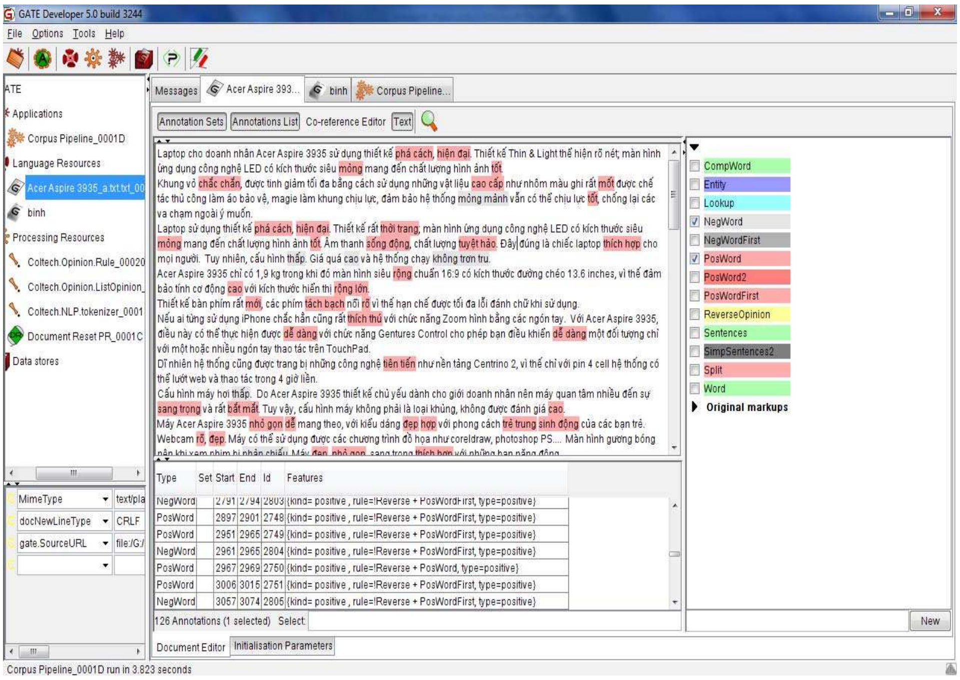

Figure 2– Sentiment words recognition in GATE