HOKKAIDO UNIVERSITY

Title	Detecting Emotive Sentences with Pattern-based Language Modelling
Author(s)	Ptaszynski, Michal; Masui, Fumito; Rzepka, Rafal; Araki, Kenji
Citation	Procedia Computer Science, 35: 484-493
Issue Date	2014-09
Doc URL	http://hdl.handle.net/2115/63685
Rights	© 2014, Elsevier. This manuscript version is made available under the CC-BY-NC-ND 3.0 license http://creativecommons.org/licenses/by-nc-nd/3.0/
Rights(URL)	http://creativecommons.org/licenses/by-nc-nd/3.0/
Type	article
File Information	1-s2.0-S1877050914010941-main.pdf

Instructions for use

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 484 – 493

Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Detecting emotive sentences with pattern-based language modelling

Michal Ptaszynski[a,*], Fumito Masui[a], Rafal Rzepka[b], Kenji Araki[b]

[a]Department of Computer Science, Kitami Institute of Technology, 165 Koen-cho, Kitami, 090-8507, Japan
[b]Graduate School of Information Science and Technology, Hokkaido University, N14 W9, Kita-ku, Sapporo, 060-0814, Japan

Abstract

This paper presents our research in detection of emotive (emotionally loaded) sentences. The task is defined as a text classification problem with an assumption that emotive sentences stand out both lexically and grammatically. The assumption is verified experimentally. The experiment is based on n-grams as well as more sophisticated patterns with disjointed elements. To deal with the sophisticated patterns a novel language modelling algorithm based on the idea of language combinatorics is applied. The results of experiments are explained with the standard means of Precision, Recall and balanced F-score. The algorithm also provides a refined list of most frequent sophisticated patterns typical for both emotive and non-emotive context.
© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Peer-review under responsibility of KES International.
Keywords: Brute Force Search Algorithms ; Pattern Extraction ; Language Modelling ; Emotive expressions ; Language Combinatorics ;

1. Introduction

Among recent developments in Natural Language Processing (NLP) research the one that has attracted increasing interest has been in the field of sentiment analysis. The goal of such research is to distinguish between sentences loaded with positive and negative attitudes. Unfortunately, the task more generic, namely, discriminating whether a sentence is even loaded with any emotional content or not, has been a topic of only few research.

In this research we decided to tackle the problem in a standardized and systematic way. We defined emotionally loaded sentences as those which in linguistics are fulfilling the emotive function of language. We also assumed that the emotive function of language is realized with various sophisticated patterns which repetitively appear in language, and that there are unique patterns for emotive sentences with comparison to non-emotive sentences. We performed experiments using a novel language modelling algorithm based on the idea of language combinatorics. By using this method we were able to minimize human effort and achieve F-score comparable to the state of the art while achieving much higher Recall rate.

The outline of the paper is as follows. Firstly, we present the background for this research and define the problem in Section 2. We also present a general literature review discussing the emotive aspects of language, and describe

*Corresponding author. Tel./fax: +81-157-26-9327.
E-mail address: ptaszynski@cs.kitami-it.ac.jp

1877-0509 © 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.129

particular previous research which try to deal with the problem of discriminating between emotive and non-emotive sentences. Section 3 describes the language combinatorics approach which we used to compare emotive and non-emotive sentences. In section 4 we describe our dataset and experiment settings. The results of the experiments as well as discussion are presented in Section 5. Finally the paper is concluded in Section 6 with several remarks regarding future work.

2. Background

There are different linguistic and paralinguistic means used to inform interlocutors of emotional states in an everyday communication. Classical linguistics distinguishes several means used particularly to express the emotional, or "emotive" meaning. These include such verbal and lexical means as exclamations[3,13], hypocoristics (endearments)[8], vulgar language[5] or, for example in Japanese, mimetic expressions (*gitaigo*)[2]. These might appear in sentences separately, or in combinations. However, when they appear the recipient (reader/listener) is immediately informed that the speaker/writer has produced their sentence in some kind of emotional state. What exactly was the state might sometimes be ambiguous and context-dependent, but the fact that something emotionally loaded has been conveyed is unquestionable.

The function of language gathering the knowledge about the above emotive elements is called the **emotive function of language**. It was first distinguished by Bühler in his *Sprachtheorie*[4] as one of three basic functions of language[1]. Bühler's theory was picked up further by Jakobson[7], who distinguished three other functions providing the basis for structural linguistics and communication studies.

To grasp the general view on how emotive meaning is realized within language, we performed a literature review on the general subject of studying emotions from the linguistic, socio-linguistic and cognitive linguistic perspective. The summary of this literature review is presented in the section 2.1.

2.1. Literature review

Research on emotions from a linguistic point of view, although still a young discipline, has already been done to some extent. For example, works of Wierzbicka[23] mark out a fresh track in research on cognitive linguistics of emotions among different cultures. Fussell[6] approached emotions from a wide cross-disciplinary perspective, trying to investigate the emotion phenomena form three broad areas: background theory of emotions, figurative language use, and social/cultural aspects of emotional communication. Weigand[22] tried to formulate a model of emotions in dialogic interactions proposing an attempt to explain emotions from the point of view of communication research. As for the Japanese language, which this research focuses on, Ptaszynski[15], made an attempt to explain both communicative and semiotic functions of emotive expressions, with a specific focus on expressions in Japanese.

Apart from research generalizing about emotions, there is also a wide range of study in the expressions of particular emotion types, or specific expressions of emotions. As for the former, a lifetime work in lexicography performed by Nakamura[11] resulted in the creation of a dictionary devoted particularly to the expressions describing states of emotions in Japanese. As for the latter, for example, Baba[2] studied Japanese mimetics in spoken discourse, Ono[13] studied emphatic functions of Japanese particle *-da*, and Sasai[19] examined *nanto*-type exclamatory sentences.

Unfortunately, there have been little linguistic research on more sophisticated emotive patterns in language. For example, a sentence "Oh, what a pleasant whether it is today, isn't it?" contains such emotive elements as interjection "oh", exclamatory sentence marker "what a", and emotive interrogative phrase "isn't it?". However, these emotive elements should rather be perceived as one pattern, like "oh, what a * isn't it?" (we discuss this pattern in more detail in section 2.3). In fact, this is one of the typical patterns of *wh*-type exclamative sentences[3]. However, although there are linguistic works investigating such emotive patterns, there has been no research experimentally confirming the existence of such patterns, or attempts to systematically and automatically extract them from larger text collections. The lack of such research is most likely caused by the limitation of typical linguistic approach in which the analysis is usually performed manually. A great help here could be offered by computer supported corpora analysis.

[1] The other two functions being *descriptive* and *impressive*.

There has been a number of research in Computational Linguistics (CL) and Natural Language Processing (NLP) focusing on the task of recognizing whether a sentence is emotive or not. We describe them in section 2.2.

2.2. Previous research

The task of recognizing whether a sentence is loaded with emotional connotation has been undertaken by a number of researchers. Rarely the task has been undertaken separately. Most often it has been performed as an additional sub-task in either sentiment analysis (SA) or affect analysis (AA) tasks. SA, in great simplification, focuses on discriminating whether a sentence conveys positive or negative attitude. AA on the other hand focuses on specifying which exactly emotion type (e.g., fear, anger, joy, etc.) has been conveyed. The fact, that the task was in most cases undertaken as a subtask of a different task, influences the way it was formulated. Below we present some of the most influential works on the topic, each formulating it in slightly different terms.

Emotional vs neutral:. Discriminating whether something is emotional or neutral is to answer a question on whether the speaker produced their sentence in an emotional state. This way the problem was investigated by Aman and Szpakowicz [1] or Neviarouskaya [12].

Subjective vs objective:. Discriminating between subjective and objective sentences is to say whether the speaker presented the sentence contents from a first-person-centric perspective or from no specific perspective. The research formulating the problem this way is, e.g, Wilson and Wiebe [24].

Emotive vs non-emotive:. Saying that a sentence is emotive means to specify the linguistic features of language which where used to produce a sentence uttered with emphasis. Research that formulated and tackled the problem this way was done by, e.g., Ptaszynski et al. [16]. Since this way of formulating the problem is the closest to ours we used Ptaszynski et al.'s system to compare with our method.

2.3. Problem definition

The task of discriminating between emotive and non-emotive sentences could be considered as a kind of automated text classification task, which is a standard task in NLP. Some of the approaches to text (or document) classification include Bag-of-Words (BOW) or n-gram. In the BOW model, a text or document is perceived as an unordered set of words. BOW thus disregards grammar and word order. An approach in which word order is retained is called the n-gram approach. First proposed by Shannon [21] over half a century ago, this approach perceives a given sentence as a set of n-long ordered sub-sequences of words. This allows matching the words while retaining the sentence word order. However, the n-gram approach allows only for a simple sequence matching, while disregarding the structure of the sentence. Although instead of words one could represent a sentence with parts of speech (POS), or dependency structure, the n-gram approach still does not allow extraction or matching of more sophisticated patterns than the subsequent strings of elements. An example of such a pattern, more sophisticated than n-gram, can be explained as follows. A sentence in Japanese *Kyō wa nante kimochi ii hi nanda !* (What a pleasant day it is today!) contains a pattern *nante * nanda !*[2]. Similar cases can be easily found in other languages, for instance, in English or Spanish. An exclamative sentence "Oh, she is so pretty, isn't she?", contains a pattern "Oh * is so * isn't *?". In Columbian Spanish, sentences "*¡Qué majo está carro!*" (What a nice car!) and "*¡Que majó está chica!*" (What a nice girl!) contain a common pattern "*¡Que majó está * !*" (What a nice * !). With another sentence, like "*¡Qué porquería de película!*" (What a crappy movie!) we can obtain a higher level generalization of this pattern, namely "*¡Que * !*" (What a * !), which is a typical *wh*-exclamative sentence pattern [3,14]. The existence of such patterns in language is common and well recognized. However, it is not possible to discover such subtle patterns using only n-gram approach.

In our research we aimed to contribute to dealing with the above problems. To do this we propose a method for language modelling and extracting from unrestricted text frequent sophisticated patterns. We also perform text classification with those patterns. The method is based on language combinatorics (LC) idea developed by Ptaszynski et al. [18].

[2] Equivalent of *wh*-exclamatives in English [3,19]; asterisk "*" used as a marker of disjoint elements.

3. Pattern-based language modelling method

To deal with the sophisticated patterns mentioned in section 2.3 we propose a language modeling method based on the idea of language combinatorics[18]. This idea assumes that linguistic entities, such as sentences can be perceived as bundles of ordered non-repeated combinations of elements (words, punctuation marks, etc.). Furthermore, the most frequent combinations appearing in many different sentences can be defined as patterns. This idea does not limit the meaning of a pattern to n-gram and assumes that sophisticated patterns with disjoint elements will provide better results than the usual bag-of-words or n-gram approach. Defining sentence patterns this way allows automatic extraction of such patterns by generating all ordered combinations of sentence elements and verifying their occurrences within a specified corpus.

Algorithms using combinatorial approach at first generate a massive number of combinations - potential answers to a given problem. This is the reason such algorithms are sometimes called brute-force search algorithms. Brute-force approach often faces the problem of exponential and rapid growth of function values during combinatorial manipulations. This phenomenon is known as combinatorial explosion[9]. Since this phenomenon often results in very long processing time, combinatorial approaches have often been disregarded. We assumed however, that combinatorial explosion can be dealt with on modern hardware to the extent needed in our research. Moreover, optimizing the combinatorial algorithm to the problem requirements should shorten the processing time making it advantageous in language processing task. Ptaszynski et al.[18] have already performed preliminary experiments in which they verified the amount of generated patterns with comparison to n-grams, and evaluated their validity using a generic sentence pattern extraction architecture SPEC. According to the evaluation, in language processing tasks it is not necessary to generate patterns of all lengths, since the most useful ones usually appear in the group of 2 to 5 element-long patterns.

Based on the above assumptions we propose a method for automatic extraction of frequent sentence patterns distinguishable for a corpus, and perform a sentence classification experiment by training a classifier on the extracted patterns. Firstly, ordered non-repeated combinations are generated from all elements of a sentence. In every n-element sentence there is k-number of combination clusters, such as that $1 \leq k \leq n$, where k represents all k-element combinations being a subset of n. The number of combinations generated for one k-element group of combinations is equal to binomial coefficient, represented in equation 1. In this procedure the system creates all combinations for all values of k from the range of $\{1, ..., n\}$. Therefore the number of all combinations is equal to the sum of all combinations from all k-element combination groups, like in equation 2.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1}$$

$$\sum_{k=1}^{n} \binom{n}{k} = \frac{n!}{1!(n-1)!} + \frac{n!}{2!(n-2)!} + ... + \frac{n!}{(n-1)!(n-(n-1))!} + \frac{n!}{n!(n-n)!} = 2^n - 1 \tag{2}$$

Next, all non-subsequent elements are separated with an asterisk ("*"). All patterns generated this way are used to extract frequent patterns appearing in a given corpus. Their occurrences O is used to calculate their normalized weight w_j according to equation 3. In the task presented in this paper we apply the method to distinguish between sentences containing emotive patterns and sentences containing non-emotive patterns. Therefore the normalized weight w_j is calculated as a ratio of all occurrences from one corpus O_{pos} to the sum of all occurrences in both corpora $O_{pos} + O_{neg}$. The weight of each pattern is also normalized to fit in range from +1 (representing purely emotive patterns) to -1 (representing purely non-emotive patterns). The normalization is achieved by subtracting 0.5 from the initial score and multiplying this intermediate product by 2. The score of one sentence is calculated as a sum of weights of patterns found in the sentence, like in equation 4.

$$w_j = \left(\frac{O_{pos}}{O_{pos} + O_{neg}} - 0.5\right) * 2 \tag{3}$$

$$score = \sum w_j, (1 \geq w_j \geq -1) \tag{4}$$

The weight can be later modified in several ways. Two features are important in weight calculation. A pattern is the more representative for a corpus when, firstly, the longer it is (length k), and the more often it appears in the corpus (occurrence O). Thus the weight can be modified by

- awarding length,
- awarding length and occurrence.

The list of frequent patterns generated in the process of pattern generation and extraction can be also further modified. When two collections of sentences of opposite features (such as "positive vs. negative", or "emotive vs. non-emotive") are compared, a generated list will contain patterns that appear uniquely in only one of the sides (e.g. uniquely positive patterns and uniquely negative patterns) or in both (ambiguous patterns). Therefore the pattern list can be further modified by deleting

- all ambiguous patterns (which weight is not +1 or -1, but somewhere in between),
- only those ambiguous patterns which appear in the same number on both sides (later called "zero patterns", since their normalized weight is equal to 0).

Moreover, a list of patterns will contain both the sophisticated patterns (with disjoint elements) as well as more common n-grams. Therefore the experiments could be performed on either all patterns, or n-grams only. Furthermore, if the initial collection of sentences was biased toward one of the sides (e.g., more sentences of one kind, or the sentences were longer, etc.), there will be more patterns of a certain sort. Thus to avoid bias in the results, instead of applying a rule of thumb, threshold is automatically optimized. The above settings are automatically verified in the process of evaluation (10-fold cross validation) to choose the best model. The metrics used in evaluation are standard Precision (P), Recall (R) and balanced F-score (F). Finally, to deal with the combinatorial explosion mentioned on the beginning of this section we applied two heuristic rules. In the preliminary experiments Ptaszynski et al.[18] found out that the most valuable patterns in language usually contain no more than six elements. Therefore we limited the scope to $k \leq 6$. Thus the procedure of pattern generation will (1) generate up to 6-element patterns, or (2) terminate at the point where no more frequent patterns were found. A diagram of the whole system is represented on Figure 1.

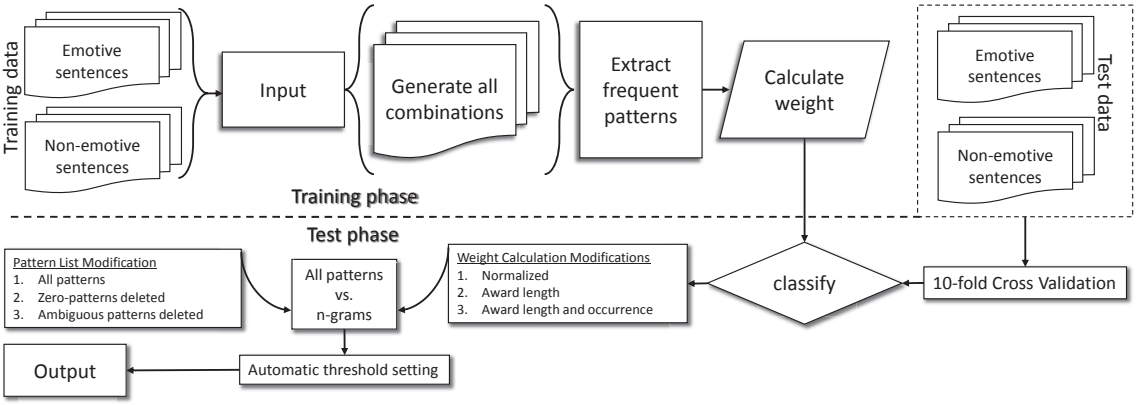


Fig. 1: Diagram of the whole system.

4. Experiments

4.1. Dataset preparation

In the experiments we used a dataset developed by Ptaszynski et al.[16] for the needs of evaluating their affect analysis system ML-Ask for Japanese language. The dataset contains 50 emotive and 41 non-emotive sentences. It was created in the following way.

Ptaszynski et al. performed an anonymous survey on thirty participants of different age and social groups. Each of them was to imagine or remember a conversation with any person or persons they know and write three sentences from that conversation: one free, one emotive, and one non-emotive. Additionally, the participants were asked to make the emotive and non-emotive sentences as close in content as possible, so the only perceivable difference was in whether a sentence was loaded with emotion or not. After that the participants also tagged the free utterances written by themselves whether or not they were emotive. Some examples from the dataset are represented in Table 1.

Table 1: Some examples from the dataset representing emotive and non-emotive sentences close in content, but differing in emotional load expressed in the sentence (Romanized Japanese / Translation).

emotive	**non-emotive**
Takasugiru kara ne / 'Cause its just too expensive	*Kōgaku na tame desu.* / Due to high cost.
Un, umai, kangeki da. / Oh, so delicious, I'm impressed.	*Kono karē wa karai.* / This curry is hot.
Nanto ano hito, kekkon suru rashii yo! / Have you heard? She's getting married!	*Ano hito ga kekkon suru rashii desu.* / They say she is gatting married.
Chō ha ga itee / Oh, how my tooth aches!	*Ha ga itai* / A tooth aches
Sugoku kirei na umi da naaa / Oh, what a beautiful sea!	*Kirei na umi desu* / This is a beautiful sea

Table 2: Three examples of preprocessing of a sentence in Japanese with and without POS tagging; N = noun, TOP = topic marker, ADV = adverbial particle, ADJ = adjective, COP = copula, INT = interjection, EXCL = exclamative mark.

Sentence example	**Preprocessing examples**
Sentence: 今日はなんて気持ちいい日なんだ！	**1. Tokens:** *Kyō wa nante kimochi ii hi nanda !*
Romanization: *Kyōwanantekimochiiihinanda!*	**2. POS:** `N TOP ADV N ADJ N COP EXCL`
Glosses: Today TOP what pleasant day COP EXCL	**3. Tokens+POS:** *Kyō*`[N]` *wa*`[TOP]` *nante*`[ADV]`
Translation: What a pleasant day it is today!	*kimochi*`[N]` *ii*`[ADJ]` *hi*`[N]` *nanda*`[COP]` *!*`[EXCL]`

The system takes as an input sentences separated into elements (words, tokens, etc.). Therefore we needed to preprocess the dataset and make the sentences separable into elements. We did this in three ways to check how the preprocessing influences the results. We used MeCab[3], a morphological analyser for Japanese to preprocess the sentences from the dataset in the three following ways:

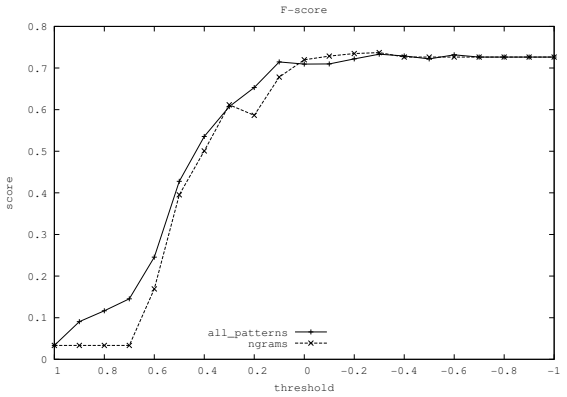
- **Tokenization:** All words, punctuation marks, etc. are separated by spaces.
- **Parts of speech (POS):** Words are replaced with their representative parts of speech.
- **Tokens with POS:** Both words and POS information is included in one element.

The examples of preprocessing are represented in Table 2. In theory, the more generalized a corpus is, the less unique patterns it will produce, but the produced patterns will be more frequent. This can be explained by comparing tokenized sentence with its POS representation. For example, in the sentence from Table 2 we can see that a simple phrase *kimochi ii* ("feeling good / pleasant") can be represented by a POS pattern `N ADJ`. We can easily assume that there will be more `N ADJ` patterns than *kimochi ii*, because many word combinations can be represented as `N ADJ`. Since there are more words in the dictionary than POS labels, the POS patterns will come in less variety but with higher occurrence frequency. By comparing the result of the classification using different preprocessing methods we can find out whether it is better to represent sentences as more generalized or as more specific.

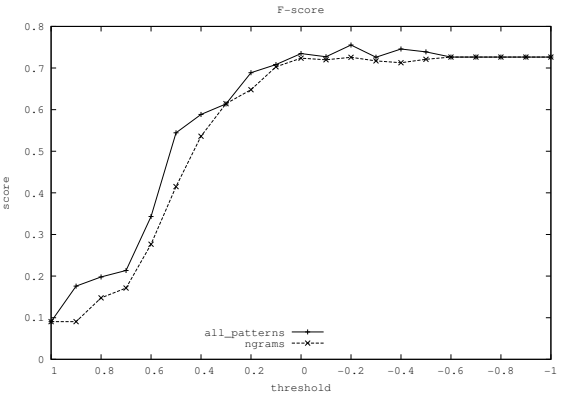
4.2. Experiment setup

The preprocessed dataset provides three separate datasets for the experiment. The experiment was performed three times, once for each kind of preprocessing. For each version of the dataset a 10-fold cross validation was performed and the results were calculated using the metrics of Precision, Recall and balanced F-score for the whole threshold span. There were two winning conditions. Firstly, we looked at which version of the algorithm achieves the top score within the threshold span. However, theoretically, an algorithm could achieve its best score for only one certain threshold, while for others it could perform poorly. Therefore we also wanted to know which version of the algorithm achieves the highest score for the longest threshold span. We calculated this as a sum of scores for all thresholds. This shows whether algorithm is balanced within the threshold span. Finally, we checked the statistical significance of the results. We used paired *t*-test because the classification results could represent only one of two values (emotive or non-emotive). To chose the best version of the algorithm we compared the results achieved by each group of modifications: "different pattern weight calculations", "pattern list modifications" and "patterns vs n-grams". We also compared the performance to the state-of-the-art, namely the affect analysis system ML-Ask developed by Ptaszynski et al.[16].

[3] https://code.google.com/p/mecab/



(a) F-score comparison between n-grams and patterns for tokenized dataset (p = 0.0209).



(b) F-score comparison between n-grams and patterns for dataset including tokens and POS (p = 0.001).

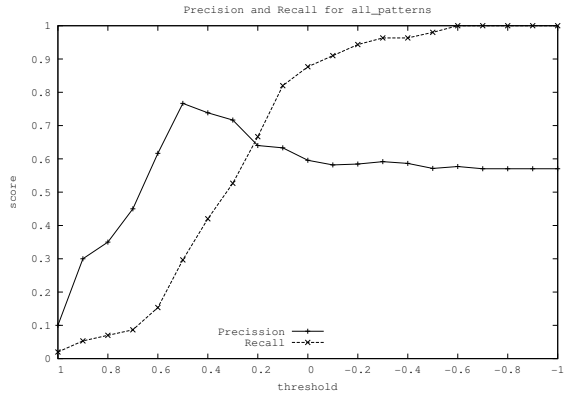
Fig. 2: An example of F-score comparison between n-grams and patterns for two datasets (tokenized only and tokens with POS with calculated statistical significance (p-value).

5. Results and discussion

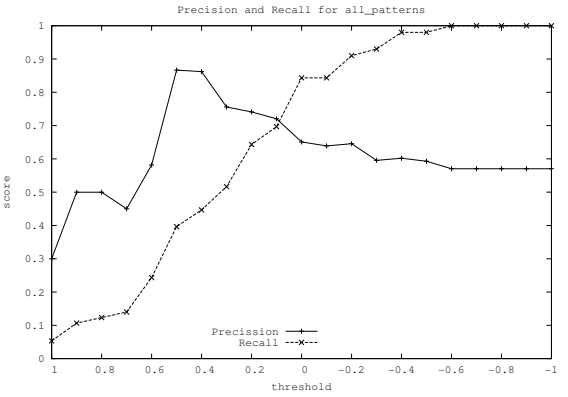
One of the main questions when using the language combinatorics approach is whether it is even necessary to use the sophisticated patterns in classification. It could happen that it is equally effective to use the usual n-gram based approach. Moreover, if the n-gram based approach was sufficient, it would be not only equally good, but also advisable to reject the combinatorial approach, since the processing time needed to learn all patterns is incomparably longer.

At first we checked the version of the algorithm using only tokenized sentences. The F-score results for tokenized sentences were not unequivocal. Usually for higher thresholds patterns achieved higher scores, while for lower thresholds the results were similar, or n-grams scored higher than patterns. Interestingly, in all situations where n-grams achieved visibly better results, the differences in results were not statistically significant. The scores, when significant, were significant on 5% level ($p<0.05$). The highest score was F = 0.75 with P = 0.61 and R = 1 for n-grams, and F = 0.74 with P = 0.6 and R = 0.96 for patterns. The algorithm usually reached its optimal F-score around 0.73–0.74. An example of F-score comparison between n-grams and patterns is represented in Figure 2a. When it comes to Precision, there always was at least one threshold for which n-grams achieved better Precision score than patterns. On the other hand, the Precision scores for patterns were more balanced, starting with a high score and slowly decreasing with the threshold span (from 1 to -1), while for n-grams, although they did achieve better results for several thresholds, they always started from a lower position and for lower thresholds more-less equalled their scores with patterns. Recall scores were better for patterns within most of the threshold span with results equalling while the threshold decreases. However, the differences were not evident and rarely statistically significant.

Next, we verified the performance using sentences preprocessed to represent POS information (nouns, verbs, etc.). In theory this type of preprocessing should provide more generalized patterns than tokens, with smaller number of patterns but with higher occurrence frequencies. Interestingly, F-scores for the algorithm with POS-preprocessed sentences revealed less constancy then for tokens. For most cases n-grams scored higher than patterns, but almost none of the results reached statistical significance. The highest F-scores were F = 0.77 with P = 0.88, and R = 0.68 for n-grams, and F = 0.74 with P = 0.59 and R = 1 for patterns. Similarly to tokens, the algorithm was usually optimized at F-score around 0.73–0.74. Slightly lower scores for patterns in this case could suggest that the algorithm itself works better with less abstracted, more specific preprocessing. Results for Precision were ambiguous. For some versions of the algorithm (e.g., unmodified, zero pattern deleted) it was better for patterns, while for others (e.g., length awarded) n-grams scored higher. The highest achieved Precision for patterns was 0.72, while for n-grams 0.71. Results for Recall confirm the results for tokens. Patterns achieved significantly higher Recall across the board.



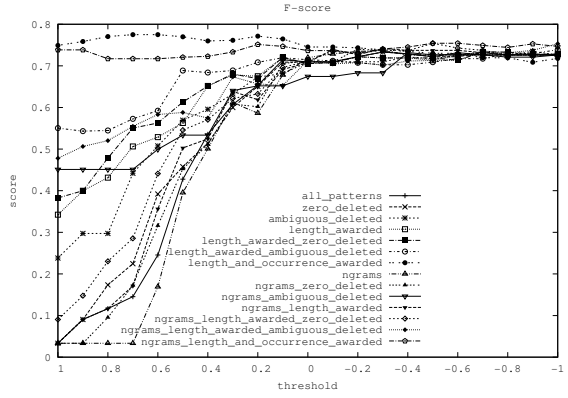
(a) Both Precision and Recall with break-even point (BEP) for the F-score (all_patterns) from Figure 2a.



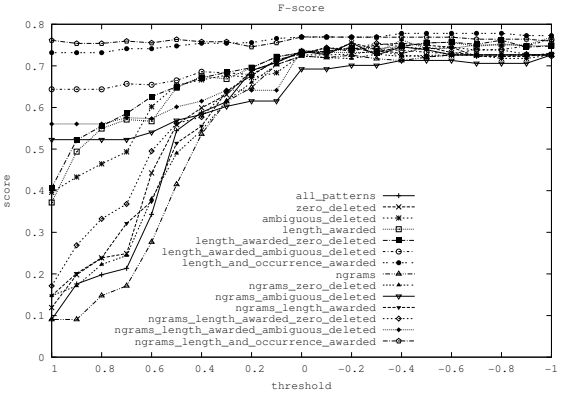
(b) Both Precision and Recall with break-even point (BEP) for the F-score (all_patterns) from Figure 2b.

Fig. 3: Both Precision and Recall with break-even points (BEP) for the F-score (all_patterns) for two datasets (tokenized and tokens with POS).

Next we used sentences preprocessed so they included both tokens and POS information. While in the previous preprocessing the elements were more abstracted (POS), the token-POS preprocessing makes the elements more specific, thus allowing extraction of a larger number, but less frequent patterns. For almost all cases the pattern-based approach achieved significantly better results, with the difference between n-grams and patterns being usually very- or extremely significant (p-value <0.01 or <0.001, respectively). The highest results for F-score were F = 0.76, with P = 0.91 and R = 0.65, or P = 0.95 and R = 0.64. The algorithm was usually reaching its optimal values around 0.75–0.76. An example of F-score comparison between n-grams and patterns is represented in Figure 2b. An comparison of F-scores for all experiment settings for two datasets (tokenized and tokens with POS) are represented in Figures 4a and 4b. An additional graph showing both Precision and Recall with the break-even point (BEP) for this F-score is represented in Figure 3b. The results for Precision were not as straightforward as for F-score. For many cases patterns scored higher, but not for the whole threshold span. However, the highest Precision was achieved by patterns with P = 0.87 for R = 0.50. Recall was usually better for patterns with the scores getting closer as the threshold decreases.



(a) F-score comparison for all experiment settings for tokenized dataset.



(b) F-score comparison for all experiment settings for dataset with tokens and POS.

Fig. 4: F-score comparison for all experiment settings for two datasets (tokenized and tokens with POS).

Table 3: Best results for each version of the method compared with the ML-Ask system.

	ML-Ask	SPEC					
		tokenized		POS		token-POS	
		n-grams	patterns	n-grams	patterns	n-grams	patterns
Precision	0.80	0.61	0.6	0.68	0.59	0.65	0.64
Recall	0.78	1.00	0.96	0.88	1.00	0.95	0.95
F-score	0.79	0.75	0.74	0.77	0.74	0.77	0.76

The affect analysis system ML-Ask developed by Ptaszynski et al.[16] on the same dataset reached the following results. F-score = 0.79, Precision = 0.8 and Recall = 0.78. The results were generally comparable, however slightly higher for ML-Ask when it comes to general F-score and Precision. Recall was always better for the proposed method. However, ML-Ask is a system developed mostly by hand for several years and is based specifically on linguistic knowledge concerning emotive function of language. On the other hand, the proposed method is fully automatic and does not need any particular preparations. Therefore, for example when performing similar task for other languages, rather than ML-Ask it would be more efficient to use our method, since it simply learns the patterns from data, while ML-Ask would require laborious preparation of appropriate databases.

5.1. Detailed analysis of learned patterns

Within some of the most frequently appearing emotive patterns there were for example: *!* (exclamation mark), *n*yo*, *cha* (emotive verb modification), *yo* (exclamative sentence ending particle), *ga*yo*, *n*!*, *n desu*, *naa* (interjection). Some examples of sentences containing those patterns are in the examples below (patterns underlined). Interestingly, most of those patterns appear in hand-crafted databases of ML-Ask (however in single word form). This suggests that it could be possible to improve ML-Ask performance by extracting additional patterns with SPEC.

Example 1. *Megane, soko ni atta nda yo.* (The glasses were over there!)
Example 2. *Uuun, butai ga mienai yo.* (Ohh, I cannot see the stage!)
Example 3. *Aaa, onaka ga suita yo.* (Ohh, I'm so hungry)

Another advantage of our method over ML-Ask is the fact that it can mark both emotive and non-emotive elements in sentence, while ML-Ask is designed to annotate only emotive elements. Some examples of extracted patterns distinguishable for non-emotive sentences were for example: *desu, wa*desu, mashi ta, masu, te*masu*. All of them are patterns described in linguistic literature as typically non-emotive, consisting in copulas (*desu*), verb endings (*masu*, and its past form *mashi ta*). Some examples of sentences containing those patterns are in the examples below (patterns underlined).

Example 4. *Kōgaku na tame desu.* (Due to high cost.)
Example 5. *Kirei na umi desu* (This is a beautiful sea)
Example 6. *Kono hon wa totemo kowai desu.* (This book is very scary.)
Example 7. *Kyo wa yuki ga futte imasu.* (It is snowing today)

6. Conclusions and future work

We presented a method for automatic extraction of patterns from emotive sentences. We assumed emotive sentences stand out both lexically and grammatically and performed experiments to verify this assumption. In the experiments we used a set of emotive and non-emotive sentences. The patterns extracted from those sentences were applied to recognize emotionally loaded and non-emotional sentences. We applied different preprocessing techniques (tokenization, POS, token-POS) to find the best version of the algorithm.

The algorithm reached its optimal F-score around 0.73–0.74 for tokenized sentences and 0.75–0.76 for tokens with POS information. The best results were achieved by patterns with both tokens and POS reaching balanced F-score of 0.76 with Precision equal to 0.64 and Recall 0.95. Precision for patterns, when compared to n-grams, was balanced, while for n-grams, although occasionally achieving high scores, the Precision was quickly decreasing. Recall scores were almost always better for patterns within most of the threshold span. By the fact that the results for sentences represented in POS were lower than the rest, we conclude that the algorithm works better with less abstracted, more specific elements.

The results of the proposed method and the affect analysis system ML-Ask were comparable. ML-Ask achieved better Precision, but lower Recall. However, since our method is fully automatic, it would be more efficient to use it for other languages. Moreover, many of the automatically extracted patterns appear in hand-crafted databases of ML-Ask, which suggests it could be possible to improve ML-Ask performance by extracting additional patterns automatically with our method. Moreover, the method is language independent while ML-Ask has been developed only for Japanese. In the near future we plan to perform experiments on datasets in other languages, as well as on larger datasets to analyse the scalability of the algorithm.

References

1. Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *Proceedings of the 10th International Conference on Text, Speech, and Dialogue (TSD-2007)*, Lecture Notes in Computer Science (LNCS), Springer-Verlag.
2. Junko Baba. 2003. Pragmatic function of Japanese mimetics in the spoken discourse of varying emotive intensity levels. *Journal of Pragmatics*, Vol. 35, No. 12, pp. 1861-1889, Elsevier.
3. Fabian Beijer. 2002. The syntax and pragmatics of exclamations and other expressive/emotional utterances. *Working Papers in Linguistics 2*, The Dept. of English in Lund.
4. Karl Bühler. 1990. *Theory of Language. Representational Function of Language*. John Benjamins Publ. (reprint from Karl Bühler. *Sprachtheorie. Die Darstellungsfunktion der Sprache*, Ullstein, Frankfurt a. M., Berlin, Wien, 1934.)
5. David Crystal. 1989. *The Cambridge Encyclopedia of Language*. Cambridge University Press.
6. Susan R. Fussell. 2002. *The Verbal Communication of Emotions: Interdisciplinary Perspectives*. Lawrence Erlbaum Associates.
7. Roman Jakobson. 1960. Closing Statement: Linguistics and Poetics. *Style in Language*, pp.350-377, The MIT Press.
8. Takashi Kamei, Rokuro Kouno and Eiichi Chino (eds.). 1996. *The Sanseido Encyclopedia of Linguistics*, Vol. VI, Sanseido.
9. Klaus Krippendorff. 1986. Combinatorial Explosion, In: Web Dictionary of Cybernetics and Systems. Princia Cybernetica Web.
10. H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text classification using string kernels. *The Journal of Machine Learning Research*, Vol. 2, pp. 419-444.
11. Akira Nakamura. 1993. *Kanjō hyōgen jiten* [Dictionary of Emotive Expressions] (in Japanese), Tokyodo Publishing.
12. Alena Neviarouskaya, Helmut Prendinger and Mitsuru Ishizuka. 2011. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering*, Vol. 17, No. 1 (2011), pp. 95-135.
13. Hajime Ono. 2002. *An emphatic particle DA and exclamatory sentences in Japanese*. University of California, Irvine.
14. Christopher Potts and Florian Schwarz. 2008. Exclamatives and heightened emotion: Extracting pragmatic generalizations from large corpora. Ms., UMass Amherst.
15. Michal Ptaszynski. 2006. *Moeru gengo: Intānetto keijiban no ue no nihongo kaiwa ni okeru kanjōhyōgen no kōzō to kigōrontekikinō no bunseki – "2channeru" denshikeijiban o rei toshite –*, [Boisterous language. Analysis of structures and semiotic functions of emotive expressions in conversation on Japanese Internet bulletin board forum '2channel'] (in Japanese), M.A. Dissertation, UAM, Poznan.
16. Michal Ptaszynski, Pawel Dybala, Rafal Rzepka and Kenji Araki. 2009. Affecting Corpora: Experiments with Automatic Affect Annotation System - A Case Study of the *2channel* Forum -, In *Proceedings of The Conference of the Pacific Association for Computational Linguistics (PACLING-09)*, pp. 223-228.
17. M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, K. Araki and Y. Momouchi. 2010. In the Service of Online Order: Tackling Cyber-Bullying with Machine Learning and Affect Analysis. *International Journal of Computational Linguistics Research*, Vol. 1 , Issue 3, pp. 135-154.
18. Michal Ptaszynski, Rafal Rzepka, Kenji Araki and Yoshio Momouchi. 2011. Language combinatorics: A sentence pattern extraction architecture based on combinatorial explosion. *International Journal of Computational Linguistics (IJCL)*, Vol. 2, Issue 1, pp. 24-36.
19. Kaori Sasai. 2006. The Structure of Modern Japanese Exclamatory Sentences: On the Structure of the *Nanto*-Type Sentence. *Studies in the Japanese Language*, Vol, 2, No. 1, pp. 16-31.
20. F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*. Vol. 34, No. 1, pp. 1-47.
21. C. E. Shannon. 1948. A Mathematical Theory of Communication, *The Bell System Technical Journal*, Vol. 27, pp. 379-423 (623-656), 1948.
22. Edda Weigand (ed.). 2002. *Emotion in Dialogic Interaction: Advances in the Complex*. John Benjamins.
23. Anna Wierzbicka. 1999. *Emotions Across Languages and Cultures: Diversity and Universals*. Cambridge University Press.
24. Theresa Wilson and Janyce Wiebe. 2005. Annotating Attributions and Private States. *Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II*, pp. 53-60.