TECHNICAL TRANSACTIONS 11/2017
CZASOPISMO TECHNICZNE 11/2017
MECHANICS
DOI: 10.4467/2353737XCT.17.200.7429

Michal Ptaszynski (ptaszynski@cs.kitami-it.ac.jp)
Fumito Masui
Department of Computer Science, Kitami Institute of Technology

Experiments with language combinatorics in text classification: lessons learned and future implications

Eksperymenty z zastosowaniem kombinatoryki językowej do klasyfikacji tekstu: dotychczasowe wnioski i implikacje na przyszłość

Abstract

This paper presents a meta-analysis of experiments performed with language combinatorics (LC), a novel language model generation and feature extraction method based on combinatorial manipulations of sentence elements (e.g., words). Along recent years LC has been applied to a number of text classification tasks, such as affect analysis, cyberbullying detection or future reference extraction. We summarize two of the most extensive experiments and discuss general implications for future implementations of combinatorial language model.

Keywords: language combinatorics, natural language processing, text classification

Streszczenie

W niniejszym artykule przedstawiono metaanalizę badań przeprowadzonych za pomocą kombinatoryki językowej (language combinatorics, LC), nowej metody generacji modelu języka i ekstrakcji cech, opartej o kombinacyjne manipulacje na elementach zdań (np. słowa). W trakcie ostatnich lat LC została zastosowana do wielu zadań z dziedziny klasyfikacji tekstu, takich jak analiza afektu, wykrywanie cyberagresji lub ekstrakcja odniesień do przyszłych wydarzeń. W niniejszym artykule podsumowujemy dwa z najbardziej obszernych doświadczeń i omawiamy ogólne implikacje dotyczące przyszłych zastosowań kombinatoryjnego modelu języka.

Słowa kluczowe: kombinatoryka językowa, przetwarzanie języków naturalnych, klasyfikacja tekstu

1. Introduction

Language modeling refers to a set of basic techniques in Natural Language Processing (NLP). It is crucial to most of NLP applications, including final word prediction [4], language identification [5], information retrieval [6], speech recognition [7], machine translation [8], part-of-speech (POS) tagging [9], or sentiment analysis [10].

However, despite such a wide applicability, there has been little progress within the language modeling field itself. There exist two to three general paradigms for language modeling, while most of the research still applies only the most basic ones, such as bag-of-words (BoW) model. Although modifications and some more sophisticated methods have been proposed (e.g., the skip-gram model), they too are bound with constraints hindering the thorough analysis of language phenomena. The more sophisticated methods for language modeling still represent a niche and are yet to be used more widely.

In this paper we analyze one of such methods called Language Combinatorics (LC) [11]. It addresses the limitations of previous models by defining a language pattern as any frequently appearing ordered combination of sentence elements. This flexible definition allows extracting from sentences all possible patterns, not limited to single words, as in the BoW model, or phrases, as in the *n*-gram model, but extends the extraction to sophisticated patterns with disjointed elements. To prove the advantage of the model, during recent several years we have extensively applied it to various experiments. This paper presents a meta-analysis of findings we drew from some of them.

The outline of the paper is as follows. We describe other research related to ours (section 2), and explain the general idea of combinatorial language model (section 3). Next we summarize (section 4) and analyze the experiments in which the model has been applied (section 5), draw general conclusions, and discuss future applications (section 6).

2. Related Research

The computationally simplest language model, the bag-of-words (BOW) model [12], considers a piece of text or document as an unordered collection of words, thus disregarding grammar and word order. Some researchers proposed improvements to BoW, e.g., by using semantic concepts instead of words (bag-of-concepts) [13], or adding word positions in sentences to the equation, thus retaining general information on word order (positional language model) [14]. Unfortunately, though one could use any general feature type to build a language model (e.g., concepts, parts-of-speech), order and longer element strings (e.g., phrases) will still be disregarded. Moreover, sentences can be of different length and any word can be preceded by another word. Thus the position of a word in sentence is not a constant value and makes the model strictly data-dependent and of limited practical use.

An approach retaining word order, based on n-grams [15], perceives an input (e.g., sentence) as a set of n-long ordered sub-sequences of elements (letters, words). Although retaining word order, n-grams allow only for simple sequence matching, while disregarding

deeper sentence structure. Again, instead of words one could use sequences of POS or concepts, however, ngrams still cannot cover more sophisticated patterns than word sequences.

An example of a language model aimed to go beyond BoW and n-grams, the skip-gram model (also known as skipped or distanced n-gram) [16], assumes that some words within an n-gram could be skipped over. In theory, this should allow extraction of most language patterns. However, to limit computational complexity of the model, skip-grams include a number of assumptions hindering the model, for example, that 1) skip-grams are generated from n-grams, not from the whole sentence, 2) a skip can appear only in one place, 3) the number of skipped elements is recorded separately for each gap (thus, two words separated by one word in between (1-skip-bigram) and by five words (5-skip-bigram) would necessarily be considered as different patterns), or that 4) the skip-length is predetermined, meaning that if the researcher chooses to extract only bigrams with only one skip in between, the same pattern, but with five skips will be disregarded from the beginning.

The above assumptions are counter-intuitive, since one can easily imagine the same sentence pattern appearing in two sentences of different length, or separated by gaps of different sizes. To illustrate this problem in Table 1 we compared which of the above-mentioned language models would discover particular patterns present in the two following sentences. The last right column represents a language model based on Language Combinatorics (LC).

1) John went to school today.
2) John went to this awful place, generously called by many school, not yesterday, but today.

Table 1. Comparison of capabilities of different language models to capture (○) or not (×) certain patterns from the corpus containing two sentences, (1) and (2)

pattern	language model			
	BoW	n-gram	skip-gram	LC
John	○	○	○	○
John went	×	○	○	○
John * to	×	×	○	○
John * school	×	×	×	○
John * to * today	×	×	×	○

Finally, in all previous research on skip-grams the model was studied only for up to 4-elements [17]. Only recent attempts used 5-element-long skip-grams [18], however, still generating them from n-grams, not the whole sentences.

Language Combinatorics is capable of dealing with any of the sophisticated patterns, by defining a pattern, or specifically, a sentence pattern, as any **ordered combination of sentence elements frequently occurring in a corpus**. This definition allows extraction of all possible meaningful linguistic patterns from unrestricted text. In our research so far, we have focused on applications of the method to various tasks from the areas of automatic pattern extraction and text classification.

3. Combinatorics-Based Language Modelling

Example: What a nice day !				
5-el. pattern:	4-el. patterns:	3-el. patterns:	2-el. patterns:	1-el. patterns:
What a nice day !	What a nice * !	a nice * !	What a	What
	What a nice day *	What a nice	What * !	a
	What a * day !	What a * !	nice * !	nice
no. of patterns: (1)	(5)	(10)	(10)	(5)

Fig. 1. Examples of various length (= number of elements) combinations extracted from one sentence

The text classification method applying combinatorial language modeling is composed of four steps: feature extraction, weight calculation, classification, and threshold optimization.

3.1. Feature Extraction with Language Combinatorics

To extract sentence patterns LC perceives sentences as bundles of ordered combinations of elements (words, etc.), and frequent combinations appearing in many sentences are defined as sentence patterns. As long as patterns are defined this way, they can be automatically extracted by generating all ordered combinations of such elements, verifying their occurrences within a corpus, and filtering out those combinations which appear only once.

In particular, in every n-element sentence there is k-number of combination clusters, such that $1 \leq k \leq n$. The number of k-element combinations is equal to binomial coefficient, represented in eq. 1. Here, all combinations for all values of k from the range of $\{1, ..., n\}$ are generated. Thus the number of all combinations generated for n-long sentence is equal to the sum of combinations from all k-element clusters of combinations, like in eq. 2.

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

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

Moreover, all non-subsequent elements are separated with an asterisk ("*"), indicating some elements appeared between those two elements. Some examples of combinations extracted this way is represented in Figure 1.

3.2. Weight Calculation

After combinations are extracted, their occurrences O are calculated. Those combinations which appeared only once are discarded, and those which appeared more than once are considered as patterns j characteristic to the sentence collection from which they were

extracted. In research applying LC in binary text classification [2, 3], the occurrences were also calculated separately for the positive side O_{pos} and the negative side O_{neg}. Such occurrences of each pattern j are further used to calculate normalized weight w_j according to equation 3, fitting the weight in the range {1, ... , -1}.

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

For purposes of a text classification experiment, previous research also modified the weight in several ways, considering what makes a pattern representative for a corpus. In particular, w_j was modified by multiplying it with:

- pattern length k_j, which provides a weight with awarded length w_{LA}, like in equation 4, or
- k_j and overall pattern occurrence (O_{pos}+O_{neg}), which provides a weight with awarded length and occurrence w_{LOA}, like in equation 5.

$$w_{LOA} = w_j * k_j * \left(O_{pos} + O_{neg} \right) \tag{4}$$

$$w_{LOA} = w_j * k_j * \left(O_{pos} + O_{neg} \right) \tag{5}$$

Moreover, when two collections of sentences of opposite features (such as "positive vs. negative") are compared, the generated pattern list will contain patterns appearing uniquely in only one of the sides or in both (later: *ambiguous patterns*, or **AMB**). A special type of an ambiguous pattern is the one appearing on both sides with the same occurrence, making its weight equal 0 (*zero-patterns*, or **0P**). Thus the list of originally generated patterns can be further modified by discarding either ambiguous patterns or zero-patterns.

3.3. Classification

In the classification, previous research applied a classifier function defined as a sum of weights of patterns found in a sentence (eq. 6).

$$\text{score} = \sum w_j, \left(1 \geq w_j \geq -1 \right) \tag{6}$$

It produces a score for each analyzed sentence. The score alone does not yet specify a class (e.g., positive or negative). The intuition suggests that the higher above or below zero is the score, the more it resembles a style of writing usually found in one of the sides. However, an intuitive rule of thumb, with zero as a universal threshold does not apply to pattern-based method, since even one word difference in a sentence can produce much larger number of patterns on one of the sides, causing an imbalance in the data. Therefore a threshold optimization of the classifier is performed to specify which threshold is optimal for the classification of provided data.

3.4. Threshold Optimization and Heuristic Rules

Data is never ideally balanced. The collections of sentences are usually biased toward one of the sides (more sentences on one of the sides, or sentences are longer, etc.). This could produce more patterns for one of the sides. To minimize the bias, instead of applying a fixed rule of thumb, a more effective way is to automatically optimize the threshold, by verifying the performance for each step and selecting the optimal one for the given data.

Finally, to deal with combinatorial explosion occurring during exhaustive combinatorial manipulations, two heuristic rules were applied. The procedure of pattern generation would (1) generate up to six elements patterns, or (2) terminate at the point where no more frequent patterns were found.

4. Applications

During recent years, LC was applied in a system for the support of experiments in text classification [19] and was used for a number of research in binary text classification. We summarize some of them below.

One of the research analyzed a small set of emotive (emotionally loaded) and non-emotive sentences. Ptaszynski et al. [3] performed a study of such sentences with the use of LC and found out that completely automatic approach to extraction of emotional patterns from sentences can give similarly good results to state-of-the-art tools developed manually and much better results than traditional classifiers (SVM).

The capability of Language Combinatorics to capture hidden patterns in language confirmed in the above research encouraged Ptaszynski et al. [2] to extract patterns of cyberbullying or Internet harassment. They analyzed a medium sized dataset containing such harmful entries, and confirmed that the LC-based method outperformed all compared previous methods for cyberbullying detection.

In another research Nakajima et al. [20] applied LC in analysis of future related expressions for trend prediction. The experiments showed that sentences referring to the future contain frequent patterns, while patterns in other sentences (present, past or other) are sparse. This proved that future-referring sentences can be analyzed as one separate kind of sentences.

5. Meta-Analysis of Experiment Results

Below we present meta-analysis of the results performed in previous papers. In the analysis we applied two datasets from the ones mentioned in section 4, namely, 1) small dataset containing emotive sentences, and 2) medium-sized dataset containing cyberbullying. We omitted the largest dataset [20] since experiments with it were performed only with one type of dataset preprocessing, while others used several kinds of preprocessing.

5.1. Datasets – Short Description

Emotive Sentences. Dataset used in [3] consists of 50 emotive and 41 non-emotive sentences collected originally by Ptaszynski et al. (2009) [21] for the needs of evaluating their affect analysis system. To collect the data they 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 they knew and write three sentences from that conversation: one free, one emotive (emotionally loaded), and one non-emotive (neutral, or non-emotional). 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 emotional load.

Cyberbullying. The dataset used in cyberbullying detection [2] contains 1,490 harmful and 1,508 non-harmful entries. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan (later: Human Rights Center) [22] and contains data from a number of informal school Websites from Mie Prefecture, Japan. The harmful and non-harmful sentences were manually labeled by experts, members of Internet Patrol, according to instructions included in an official governmental manual for dealing with cyberbullying [23].

5.2. General Setup of Experiments

Both analyzed research used similar experiment setup. The prepared datasets were used in a text classification experiment with the use of the proposed LC-based method, and other methods (previously developed systems and classifiers). In classification, researchers compared the performance of sophisticated patterns to more common n-grams, and BoW. Feature weights were calculated according to the equations explained in section 3.2. For classifiers based on BoW, a traditional weight calculation scheme was also applied, namely, term frequency (*tf*), and term frequency multiplied by inverted document frequency (*tf*idf*).

Dataset Preprocessing. Both datasets were in Japanese and were preprocessed in the following ways (• – features used in both experiments; ◦ – features used only in cyberbullying detection).

- **Tokenization**: All words, punctuation marks, etc. are treated as separate features (later: **`TOK`**).
- **Lemmatization**: Same as above but the words are represented in their generic (dictionary) forms, or "lemmas" (later: **`LEM`**).
- **Parts of speech (POS)**: POS are used instead of words (later: **`POS`**).
- **Tokens with POS**: Both words and POS information is included in one element (later: **`TOK+POS`**).
- **Lemmas with POS**: Same as above but with lemmas instead of words (later: **`LEM+POS`**).
- **Chunking**: Sentences are divided into sub-parts (chunks) by grammatical rules, such as noun phrases, verb phrases, etc. (later: **`CHUNK`**).

- **Dependency structure**: Same as above, but with information on grammatical relations between chunks (later: **DEP**).
- **Chunks with Named Entities**: Chunks with named entities (private names, numericals, etc.) annotated on sentences (later: **CHUNK+NER**).
- **Dependency structure with Named Entities**: Both dependency relations and named entities are used (later: **DEP+NER**).

Each kind of preprocessing (or feature set) represents a different level of generalization. Higher sentence generalization produces less unique patterns, but the produced patterns are more frequent. This can be explained by comparing a tokenized (low generalization) sentence with its POS representation (high generalization). For example, in the sentence from Figure 1 the phrase "nice day" is represented by POS as ADJ N. There will be more ADJ N patterns than nice day, because many word combinations can be represented as ADJ N. There are also more words in a dictionary (around ten thousand) than POS labels (about a dozen). Comparison of classification results for different preprocessing methods can help specify whether it is better to represent sentences as more generalized or as more specific.

In meta-analysis we re-analyzed the results of experiments to answer the following questions:

- Is LC better than simple language modeling methods (n-grams, BoW)?
- Which preprocessing method (feature set) was the best?
- Which classifier modification was the best? (see sec. 3.2)

To answer these questions we compared the highest achieved balanced F-score within the threshold span achieved by each feature set. We also checked the correlations between generalization level of features and performance of each classifier modification. We also looked at break-even points (BEP) of Precision and Recall, showing which version was more balanced.

5.3. Small Dataset: Emotive Sentences

5.3.1. F-score Comparison Between Feature Sets

The highest achieved F-score was obtained by parts-of-speech (.774) while tokenized dataset with POS scored as second (.769). Both lemmatized datasets achieved the lowest scores (.744 and .746 for lemmas alone and with POS, separately). The initial intuition would suggest that parts-of-speech were the optimal setting, while lemmatization decreased the results. Worse results also tended to have wider dispersion between Precision and Recall.

As for the performance of modifications, all of the best classifier versions always used length awarded (LA), with either all ambiguous patterns (AMB) or zero-patterns (0P) deleted from pattern lists. No straightforward answer was obtained whether it was more useful to use patterns or n-grams. Although three out of five highest-scoring settings were based on n-grams (POS, TOK+POS, TOK), patterns were always second best and the differences were not significant. The results showing the highest achieved F-scores were represented in Figure 2.

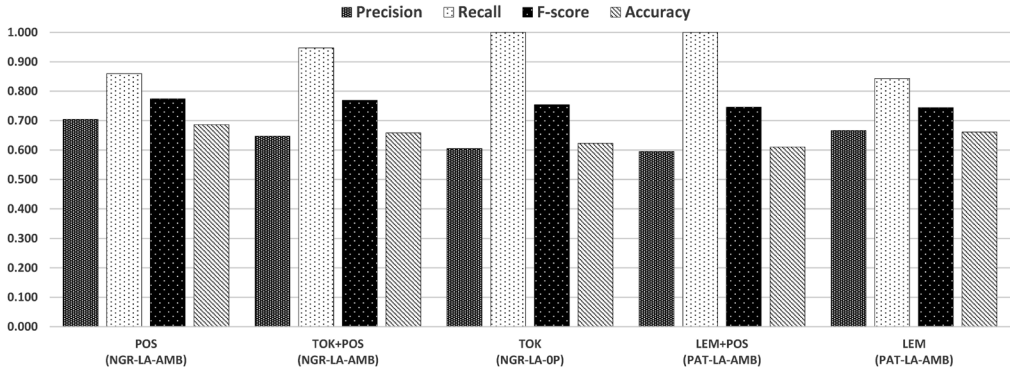

Fig. 2. Best F-scores for each preprocessing of emotive sentence dataset, ordered from left to right, with corresponding Precision, Recall and Accuracy. Classifier version that achieved the score – in brackets

Standard SVM-based classifier trained on Bag-of-Words language model and *tf*idf* weighting scored much lower, with the highest score of F1 = 73%, which indicates that simple BoW language model, even when used to train an efficient SVM classifier is not suitable for classification of emotive language.

Table 2. Comparison of Break-Even Points (BEP) of Precision and Recall for all classifier versions and preprocessing types for emotive sentences dataset. Best within each preprocessing group in bold type font. Best within each classifier type underlined

Feature sets	TOK	POS	TOK+POS	LEM	LEM+POS
PAT-ALL	0.650	0.701	0.713	0.649	0.632
PAT-0P	0.637	**0.710**	0.713	0.627	0.653
PAT-AMB	0.624	0.560	0.702	**0.664**	0.637
PAT-LA	0.679	-	0.722	0.551	0.651
PAT-LA-0P	0.676	-	0.722	0.626	0.659
PAT-LA-AMB	**0.688**	-	0.716	-	-
NGR-ALL	0.594	0.695	0.712	0.629	0.665
NGR-0P	0.609	**0.697**	0.712	0.633	0.665
NGR-AMB	0.595	0.668	0.664	0.610	0.628
NGR-LA	0.620	**0.680**	**0.723**	0.635	**0.682**
NGR-LA-0P	0.633	**0.680**	**0.723**	0.645	**0.682**
NGR-LA-AMB	**0.665**	-	**0.707**	0.652	0.655

5.3.2. Break-Even Point Analysis

In the BEP analysis we looked 1) which classifier version got the highest BEP, 2) which usually got the highest BEP for different dataset preprocessing, and 3) which preprocessing most often provided highest BEP.

The comparison revealed that TOK+POS dataset almost always performed best, achieving the highest BEP. This stands somewhat in contradiction to the results for F-scores, where

POS dataset obtained highest scores. However, detailed analysis revealed that, although POS achieved the single highest F-score, for other thresholds they scored similarly, or lower than other datasets. Moreover, even with POS as the highest, TOK+POS was still the second-best. While also being the most balanced (highest BEP for almost all cases), TOK+POS could be the optimal for analysis of emotive sentences.

This would also suggest that the method works better on more specific, less generalized features. Although the best BEP of all, with P = R = F = 0.723 was achieved by n-gram based classifier awarding pattern length in weight calculation, again, there was no clear answer whether it was better to use patterns, or *n*-grams. Comparison of all BEPs for all classifier versions and experiment settings is represented in Table 2.

5.3.3. Influence of Dataset Generalization on Results

Next we analyzed the influence of dataset preprocessing on the results. To achieve this we needed a quantifiable measure showing dataset generalization. A dataset is the more generalized, the fewer number of frequently appearing unique features it produces. Therefore to estimate dataset generalization level we decided to apply Lexical Density (LD) score [24]. It is a score representing an estimated measure of content per lexical units for a given corpus, and is calculated as the number of all unique words from the corpus divided by the number of all words in the corpus. However, since in our research we used a variety of different features, not only words, we will further call this measure Feature-based Lexical Density, or shortly, Feature Density (FD).

Table 3. Analysis of influence of dataset generalization for emotive sentences dataset

Dataset Preprocessing		**No. of unique unigrams**	**No. of all unigrams**	**Feature Density (FD)**	**Highest achieved F-score**	**Highest unmodified F-score**	**BEP**
Fetaure sophistication ←low high→	**TOK+POS**	311	821	0.3788	0.769	0.755	0.723
	TOK	306	821	0.3727	0.754	0.733	0.688
	LEM+POS	280	821	0.3410	0.746	0.733	0.682
	LEM	276	821	0.3362	0.744	0.733	0.664
	POS	12	819	0.0147	0.774	0.728	0.710
		unique 1ngr with			**FD with**		
		F1	**F1-unmod.**	**BEP**	**F1**	**F1-unmod.**	**BEP**
Pearson Correlation		-0.6018	0.5114	-0.3007	-0.6018	0.5114	-0.3007
Coefficient (*p*-value)		($p = 0.283$)	($p = 0.378$)	($p = 0.623$)	($p = 0.283$)	($p = 0.378$)	($p = 0.623$)
with statistical				**F1 & BEP**		**F1-unmod. & BEP**	
significance				*0.921		0.5735	
(p-value)				($p = 0.0265$)		($p = 0.312$)	

* $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$

After calculating FD for all datasets we calculated Pearson's correlation coefficient (*p*-value) to see if there was any correlation between dataset generalization (FD) and the

results. Pearson's coefficient can achieve scores from 1.0 (perfect positive correlation), through 0.0 (no correlation) to -1.0 (perfect negative correlation). In comparison we used the highest achieved F-scores. However, since the highest overall F-scores were various classifier settings (all patterns, or zero-patterns deleted; with length awarded, or not, etc.), we also used an unmodified version of the classifier (**PAT-ALL**). As an equivalent set of results we also used BEPs. Finally, we verified whether the correlations were statistically significant.

Firstly, the highest achieved F-scores were significantly positively correlated with BEPs, which indicates that both measures show in general similar tendencies. The highest achieved F-scores indicated somewhat strong negative correlation with both unique unigrams as well as with Feature Density score, and mild negative correlation with BEPs. Interestingly, highest F-scores for unmodified dataset (all patterns) were somewhat positively correlated with unique unigrams and FD. The two results stand in contradiction, since the first one suggests that the less dense is the feature set the higher the results will get, while the second result indicates the opposite. This is most probably due to the least feature-dense POS-tagged dataset, which achieved the highest score. Unfortunately, neither of the correlations were statistically significant.

5.4. Medium-sized Dataset: Cyberbullying

5.4.1. F-score Comparison Between Feature Sets

The best F-score (.803) was achieved by lemmatization with POS information. Interestingly, while the winning settings showed high consistency between Precision and Recall, close to BEP (.802), for other preprocessing settings, the lower was the F-score, the wider was the gap between P and R. This is meaningful not only regarding the general performance, also provides insight into the influence of generalization on performance.

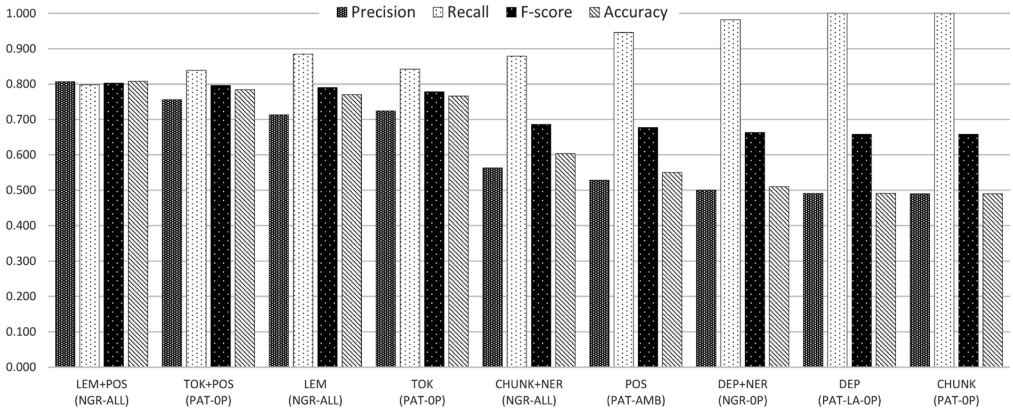

Fig. 3. Best F-scores for each dataset, ordered from left to right, with corresponding Precision, Recall and Accuracy. Classifier version that achieved the score – in brackets

Although the best score was achieved by n-grams, both settings were the highest interchangeably. For example, second best (F1 = 0.796) was pattern-based (TOK+POS/PAT-0P) third best was n-gram based, fourth – again – patterns, etc. This suggests that we need to perform more experiments, most desirably on a wider threshold span to choose whether patterns or n-grams are better. The most optimal classifier settings was the unmodified one, or the one with zero-patterns deleted. This suggests, that, in the case of cyberbullying messages, it is more effective to use ambiguous patterns in classification. The results can be clustered into two groups: with a small and with a wide gap between P and R This grouping is similar further in BEP analysis. Figure 3 shows the F-scores ordered decreasingly from left to right.

5.4.2. Break-Even Point Analysis

As for BEPs, the highest score of all (P = R = F1 = .802) was achieved by n-gram-based classifier. Lemmatized dataset combined with part-of-speech usually scored highest, differently to emotive sentence dataset, where this setting was one of the worst. The method usually performed better on more specific feature sets ((TOK, LEM, TOK+POS, LEM+POS), than for more generalized ones (POS, CHUNK, DEP, CHUNK+NER, DEP+NER). The results for best BEPs for all versions of the classifier were represented in Table 4.

Table 4. Break-even points for all feature sets on cyberbullying dataset

Feature sets	TOK	LEM	POS	TOK +POS	LEM +POS	CHUNK	DEP	CHUNK +NER	DEP +NER
PAT	0.761	0.751	0.613	0.785	0.781	**0.633**	0.566	0.603	0.510
PAT-0P	0.763	0.751	0.613	**0.786**	0.781	0.632	0.551	0.605	0.512
PAT-AMB	**0.770**	0.751	0.613	0.764	0.782	0.629	**0.591**	0.603	0.514
PAT-LA	0.729	0.748	0.613	0.726	0.781	0.632	0.568	-	0.505
PAT-LA-0P	0.729	0.737	0.596	0.726	0.760	**0.633**	0.549	-	0.505
PAT-LA-AMB	0.711	0.737	0.594	0.715	0.761	0.629	**0.591**	-	0.516
NGR	0.761	**0.784**	**0.614**	0.785	**0.802**	0.632	0.566	**0.655**	0.547
NGR-0P	0.762	**0.784**	0.613	**0.786**	**0.802**	0.632	0.551	0.652	**0.548**
NGR-AMB	**0.770**	0.767	0.570	0.764	0.777	0.612	**0.591**	0.610	0.526
NGR-LA	0.729	0.767	0.605	0.726	0.762	**0.633**	0.551	0.619	0.546
NGR-LA-0P	0.729	0.768	0.607	0.726	0.769	0.631	0.559	0.622	**0.548**
NGR-LA-AMB	0.711	0.762	0.596	0.715	0.750	0.613	0.589	0.589	0.529

5.4.3. Influence of Generalization on Results

Feature Density score revealed somewhat strong negative correlation (around –0.7) between the results and FD. This means that the results were better when the FD was low. The correlation was not ideal due to the fact that the dataset with the lowest FD (POS)

achieved one of the lowest results. Interestingly, preprocessing methods resulting in very high FD (dependency parsing, etc.) also achieved similarly low results. For the given datasets the performance is growing along with decreasing FD, until the lowest FD is reached (POS), which also obtained low results. Thus, in the future we plan to use the FD measure to find a preprocessing method with optimal feature density, resulting in even better results. The analysis of influence of dataset generalization on results is represented in Table 5.

Table 5. Analysis of influence of generalization on results for cyberbullying dataset

Dataset Preprocessing			No. of unique unigrams	No. of all unigrams	Feature Density	Highest achieved F-score	Highest unmodified F-score	BEP
Feature sophistication	←low high→	DEP	12802	13957	0.917	0.658	0.658	0.591
		DEP+NER	12160	13956	0.871	0.663	0.662	0.548
		CHUNK	11389	13960	0.816	0.658	0.658	0.633
		CHUNK+NER	10657	13872	0.768	0.686	0.684	0.655
		TOK+POS	6565	34874	0.188	0.796	0.795	0.786
		TOK	6464	36234	0.178	0.778	0.778	0.770
		LEM+POS	6227	36426	0.171	0.803	0.783	0.802
		LEM	6103	36412	0.168	0.790	0.764	0.784
		POS	13	26650	0.000	0.677	0.677	0.614
			unique 1ngr with			FD with		
			F1	F1-unmod.	BEP	F1	F1-unmod.	BEP
Pearson Correlation			-0.450	-0.453	-0.431	-0.735	-0.736	-0.706
Coefficient (*p*-value)			($p = 0.224$)	($p = 0.221$)	($p = 0.247$)	($p = 0.0242$)	($p = 0.024$)	($p = 0.0336$)
with statistical					F1 & BEP		F1-unmod. & BEP	
significance					0.9681		0.9595	
(*p*-value)					($p = 0.00002$)		($p = 0.00004$)	

* $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$

6. Conclusions and Future Work

We presented a meta-analysis of experiments performed with Language Combinatorics (LC), a novel method for language model generation based on combinatorial manipulations of sentence elements. For the analysis we selected the most exhaustive experiments, namely, emotive sentence detection (small dataset) [3] and cyberbullying detection (medium-sized dataset) [2]. We briefly summarized the experiments and discussed general implications for future implementations of LC.

Meta-analysis revealed many contradictory results. For example, POS-tagged dataset obtained the highest F-score for the small dataset, but the worst for the medium-sized dataset.

Feature Density (FD) significantly correlated with F-scores and BEPs for mid-size dataset, but not for small dataset. Also, results for pattern list and classifier modifications were not consistent among the two datasets. For small dataset awarding pattern length resulted in better scores, usually boosted further by deleting ambiguous patterns. For medium dataset such modifications usually hindered the results.

As for similarities, awarding both pattern length and occurrence in classification was usually not effective. Therefore this option could be discarded in future experiments to reduce the overall time required for experiment.

In the future we plan to unify the meta-analysis. This time small dataset also had smaller number of preprocessing variations used in experiments, which could influence the correlations of Feature Density with results. Also, although there was no clear answer on whether patterns or n-grams were more effective, both always produced better results than Bag-of-Words model. To further confirm this result, we plan to extend the scope of patterns length from six elements to the maximal possible length, since the least frequent patterns will still be filtered out during the feature extraction procedure.

We plan to train other classifiers (SVM, Neural Networks, etc.) on the proposed pattern-based language model. This however will require much stronger hardware than was available at the time of writing. Finally, the results of meta-analysis could have been influenced by various differences in datasets – not only in their sizes, but also, e.g., the type of language. Therefore in the future we plan to repeat the experiments on size-unified datasets.

References

[1] Ptaszynski M., Masui F., Rzepka R., Araki K., *First Glance on Pattern-based Language Modeling*, Language Acquisition and Understanding Research Group Technical Reports, 2014.

[2] Ptaszynski M., Masui F., Kimura Y., Rzepka R., Araki K., *Extracting Patterns of Harmful Expressions for Cyberbullying Detection*, Proceedings of LTC'15, 2016, 370-375.

[3] Ptaszynski M., Masui F., Rzepka R., Araki K., *Subjective? Emotional? Emotive?: Language Combinatorics based Automatic Detection of Emotionally Loaded Sentences*, Linguistics and Literature Studies, Vol. 5, No. 1, 2017, 36-50.

[4] Bickel S., Haider P., Scheffer T., *Predicting sentences using n-gram language models*, Proceedings of HLT-EMNLP 2005, 2005, 193-200.

[5] Li Haizhou, Bin Ma, *A phonotactic language model for spoken language identification*, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005, 515-522.

[6] Ponte J.M., Croft W.B., *A language modeling approach to information retrieval*, Proceedings of the 21st annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, 275-281.

[7] Brown P.F., Cocke J., Pietra S.A.D., Pietra V.J.D., Jelinek F., Lafferty J.D., Mercer R.L., Roossin P.S., *A statistical approach to machine translation*, Computational linguistics, Vol. 16, No. 2, 1990, 79-85.

[8] Mays E., Damerau F.J., Mercer R.L., *Context based spelling correction*, Information Processing & Management, Vol. 27, No. 5, 1991, 517-522.
[9] Kupiec J., *Robust part-of-speech tagging using a hidden Markov model*, Computer Speech & Language, Vol. 6, No.3, 1992, 225-242.
[10] Hu Y., Lu R., Li X., Chen Y., Duan J., *A language modeling approach to sentiment analysis*, Computational Science – ICCS 2007, 1186-1193.
[11] Ptaszynski M., Rzepka R., Araki K., Momouchi Y., *Language combinatorics: A sentence pattern extraction architecture based on combinatorial explosion*, International Journal of Computational Linguistics (IJCL), Vol. 2, No. 1, 2011, 24-36.
[12] Harris Z., *Distributional Structure*, Word, Vol. 10, N. 2/3, 1954, 146-162.
[13] Cambria E., Hussain A., *Sentic Computing: Techniques, Tools, and Applications*, Springer, 2012.
[14] Lu Y., Zhai C.X., *Positional Language Models for Information Retrieval*, 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, 299-306.
[15] Markov A.A., *Extension of the limit theorems of probability theory to a sum of variables connected in a chain*, Reprinted in Appendix B of: R. Howard, *Dynamic Probabilistic Systems*, Vol. 1: *Markov Chains*, John Wiley and Sons, 1971.
[16] Huang X., Alleva F., Hon H.W., Hwang M.Y., Rosenfeld R., *The SPHINX-II Speech Recognition System: An Overview*, Computer, Speech and Language, Vol. 7, 1992, 137-148.
[17] Guthrie D., Allison B., Liu W., Guthrie L., Wilks Y., *A closer look at skip-gram modelling*, Proceedings of LREC-2006, 2006, 1-4.
[18] Pickhardt R., Gottron T., Korner M., Wagner P.G., Speicher T., Staab S., *A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser Ney Smoothing*, Proceedings of ACL 2014, 2014, 1145-1154.
[19] Ptaszynski M., Lempa P., Masui F., *A Modular System for Support of Experiments in Text Classification*, Technical Transactions, vol. 7-B/2015, 229-243.
[20] Nakajima Y., Ptaszynski M., Honma H., Masui F., *Investigation of Future Reference Expressions in Trend Information*, Proceedings of the 2014 AAAI Spring Symposium Series, 2014, 31-38.
[21] Ptaszynski M., Dybala P., Rzepka R., Araki K., *Affecting Corpora: Experiments with Automatic Affect Annotation System – A Case Study of the 2channel Forum*, Proceedings of PACLING-09, 2009, 223-228.
[22] Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan, http://www.pref.mie.lg.jp/jinkenc/hp/ (access: 21.04.2017).
[23] Ministry of Education, Culture, Sports, Science and Technology (MEXT), *'Netto-jo no ijime' ni kansuru taio manyuaru jirei shu (gakko, kyoin muke)*, MEXT, 2008.
[24] Ure J., *Lexical density and register differentiation*, [in:] *Applications of Linguistics*, (eds.) G. Perren, J.L.M. Trim, Cambridge University Press, London 1971, 443-452.