ISSN 2007-9737

On Detecting Keywords for Concept Mapping in Plain Text

Juan Huetle Figueroa[1], Fernando Perez Tellez[1], David Pinto[2]

[1] Institute of Technology Tallaght,
Ireland

[2] Benemérita Universidad Autónoma de Puebla,
Faculty of Computer Science,
Mexico

{juan.huetle, fernandopt}@gmail.com, dpinto@cs.buap.mx

Abstract. The key terminology is very important for scientific works, especially for Natural Language Processing field. However, there is no optimal way to extract all the key terminology in a reliable manner. Thereby it is important to develop automatic methods for extracting key terms. This document presents a way to obtain the key terminology based on labels that were manually obtained by an expert in the area. Subsequently, we got POS (Part-of-the-speech) tags for each label, in which we obtained patterns from key terminology that were used as filters afterwards. Experiment 1 was tested using the labels obtained manually and the labels obtained by the proposed approach, with 60% of the corpus for training and 40% for tests. The patterns were evaluated with three different measures of evaluation such as precision, recall, and F-measure. Experiment 2 used three measures for ranking N-grams (sequence of terms), Point mutual information, Likelihood-ratio, and Chi-square. To obtain the best N-grams, we have implemented in experiment 3 intersections between the previous measures and filtering N-grams by POS patterns. Also, they were compared with the manually labeled set, evaluation measures were used to see its result, gave us a good recall moreover acceptable precision and F-measure. In experiment 4 POS patterns were tested in a much larger corpus of a different domain obtaining slightly higher results.

Keywords. Collocations, n-gramas, POS, keyword extraction.

1 Introduction

The key terminology refers to ”the body of terms used with a particular technical application in a subject of study, profession, etc.”[1] Within a series of files called corpus, which can be composed of a word or many words. In this research work, we have referred to as ’key N-grams’ to the key terminology of different length identified in a raw text. We have specifically worked with three types of key N-grams and we refer to them as bigrams (two keywords), trigrams (three keywords) and quadrigrams (four keywords). This key terminology is related to many research works, especially the area of natural language processing (NLP).

The importance of the experiments presented in this research work details ways to obtain key terminology and its evaluation. We used evaluation measures to validate the results obtained such as precision, recall, and F-measure. To achieve this goal, we have defined a base corpus based on labels manually provided by an expert in the IT area.

Part-of-Speech (POS) tags were extracted from each of the manually labeled text, with which were obtained patterns, that in this research work we will be referring as POS patterns. To verify the reliability of the POS patterns, they were tested with another different corpus.

[1] https://en.oxforddictionaries.com/definition/terminology

doi: 10.13053/CyS-24-2-3400

Table 1. Normal POS

Punctuation Marks: “““”, “:”	RB: adverb
“;”, “””, “(”, “)”, “,”, “–”, “.”	RBR: adverb, comparative
CC: conjunction, coordinating	RBS: adverb, superlative
CD: numeral, cardinal	RP: particle
DT: determiner	SYM: symbol
EX: existential there	TO: ”to” as preposition
FW: foreign word	VB: verb, base form
IN: preposition or conjunction	VBD: verb, past tense
JJ: adjective or numeral	VBG: verb, present
JJR: adjective, comparative	participle or gerund
JJS: adjective, superlative	VBN: verb, past participle
MD: modal auxiliary	VBP: verb, present tense,
NN: noun, common	not 3rd person singular
NNP: noun, proper, singular	VBZ: verb, present tense,
NNS: noun, common, plural	3rd person singular
PDT: pre-determiner	WDT: WH-determiner
POS: genitive marker	WP: WH-pronoun
PRP: pronoun, personal	WRB: Wh-adverb
PRP$: pronoun, possessive	

The measures used are Pointwise mutual information (PMI), Likelihood-Radio (LKH-R) and Chi-square (Chi-S). They were chosen for simplicity, low calculation capacity required and showed acceptable results in the experiments. In addition, we have defined a range of measures to compare them and detect the best key terminology in any text.

In this research, we created three different experiments. The first was for the extraction of POS patterns. They are shown in Table 1. They are forty-five possible labels offered by the NLTK toolkit. We decided that stop words must be included, because the keywords that were manually labeled include words such as *the, to, in* and *of*, an example of keywords with stop words are *the latest technologies*, *ability to lead*, *degree in statistics* and *knowledge of firewalls*.

The stop words defined important patterns such as (*NN TO NN*), (*NN IN NN*) and (*NN IN NNS*), consequently the evaluation measures such as precision, recall, and F-measure were used, showing the efficiency of POS patterns.

In the second experiment, we used the intersection between the sets of N-grams ranked by collocation measures and filtered by the highest values, in addition to including the filtering by the patterns obtained in the first experiment.

In the third experiment, the intersection task is when a set of N-grams filtered by the highest value of a collocation measure appears in another set of N-grams filtered by a different collocation measure. In the table 6 appears the phrase *Dublin city center* with a high value in PMI, Likelihood-Radio and Chi-square. For the following experiment, we used POS patterns as a filter in the intersections.

The rest of this document is organised as follows. The following section provides a review of the related work to obtain key terminology, methods, and applications. In Section 3, describes the corpus (dataset) used for the experiments. Section 4 contains a description of the pre-processing task of the data. Section 5 we describe the measures used and the intersection of the terms with their measurements and the experiments carried out in this research work. Section 6 provided an evaluation and comparison of the results obtained. Finally, in the last section, we conclude the document and outlines the future work directions.

2 Related Work

Our research goal is to obtain key terminology from plain documents, we have studied previous research works focused on keyword extraction. Researchers in [18] have reported the use of statistical methods and approaches such as simple statistics, linguistics, machine learning approaches. They extracted a small set of units, composed of one or more terms, from a single document. They discussed the extraction of small units sets, composed of one or more terms, from a single document. It is an important problem in Text Mining (TM), Information Retrieval (IR) and Natural Language Processing (NLP). Authors focused on the graph-based methods.

They have compared methods with existing supervised and unsupervised methods. On the other hand, [9] used statistical methods with TF-IDF (term frequency, inverse document frequency) they described the use of TF-IDF in different parts of the plain document. For example, if a word appears sporadically in more than half of the document it is also considered as a keyword without taking into account the stop words.

As well as multiple times in a single paragraph but not in the overall document TF-IDF will not consider the word as keyword considering its low frequency.

In [8] authors used unsupervised approaches to automate the keyword extraction process from meeting transcript documents and they incorporated the use part-of-speech (POS) information in a similar manner that we did. Then, they identified key-words using F-measure and a weighted score relative, giving them good results with TFIDF. The data that they used was *meeting recordings* converted into text. A different research work [17] using the knowledge graph that combines semantic similarity clustering algorithms show good results using evaluation measures such as precision, recall, and F-measure. They adopted in previous research works the syntactic rule *(JJ)*(NN—NNS—NNP—NNPS)+*, where * and + mean zero or more adjectives, giving them good results.

Another unsupervised keyphrase extraction is [7] the authors used four public corpora to demonstrate that they proposal improved the performance of keyphrase extraction. They demonstrated that to use participles, adverbs and cardinal numbers is better at extracting keyphrase that only use adjectives and noun. They introduced two methods to remove unnecessary labels:

— **First method:** Begins with a POS label such as: *JJ, JJR, JJS, NN, NNS, NNP,* or *NNPS*; and ends with a POS tag such as: *NN, NNS, NNP* OR *NNPS*.

— **Second method:** Begins with a POS label such as: *JJ, JJR, JJS, NN, NNS, NNP,* or *NNPS*; and ends with a POS tag such as: *NN, NNS, NNP* OR *NNPS*.

One novel way to extract key phases is the research work [12] where the authors used a semantic relationship graph. They archive an improvement of 5.3% and 7.3% over keyphrases used in the evaluation *SemEval-2010*. For they tagging documents they used Stanford Log-linear POS Tagger. Their method is less restrictive used labels such as *NN, NNS, NNP, NNP, NNPS, JJ, JJR*, and *JJS*.

The authors of [14] automatically generated a headline for a single document. They mixed sentence extraction and machine learning, their corpus were scientific articles. Another interesting approach is [1] they combine resources for lexical analysis such as an electronic dictionary, tree tagger, WordNet, N-grams, POS pattern, resulting in a survey, they used different dataset the most relevant for us is the web pages, encyclopedia article, newspaper articles, journal articles, and technical report. In [19] used *salience* rank in 500 news articles, the result was to improve the quality of extracted keyphrases and balance topic in the corpus.

There is also some research in the field of real-time automatic speech recognition. In [4] authors applied keywords to formulate implicit queries to a just-in-time-retrieval system for use in meeting rooms.

3 Dataset

In this research work, we were working with job descriptions, all the data was taken from jobs.ie[2] a website in Ireland. The website has 46 different categories (some relevant examples are in Table 2) and 6,917 jobs description at the moment of writing this paper. Each job description file contains information such as skills needed, payments and area of work. All the documents were in HTML and JSON format, we cleaned the documents from HTML tags, and download the updated information for each week.

For this research work, we used in specific the IT (information technology) list count with 153 jobs descriptions, the average per file is 3 kilobytes. The IT list was chosen because we have an expert in that field who extracted keywords manually required to validate the results, for future work, it is intended to obtain experts in other areas.

To collect these data we used a web crawler (HTTrack)[3] to automatically download all the jobs descriptions every week.

The reasons to chose these data are:

[2] https://www.jobs.ie/

[3] https://www.httrack.com

Table 2. Example of job description categories

Sector	Fq	Sector	Fq
Hotels	1021	Manager/Supervisor	267
Restaurant/Catering	669	Secre./Admin/PA	257
Chef Jobs	374	Pubs/Bars/Clubs	199
Call-Centre/Serv.	340	Health/Med./Nursing	156
Accountancy/Finance	304	**IT**	153
Sales - Up to 35k	297	WH./Logis./Ship.	153
Retail	293	Trades/Operative	144
Sales - 35k+	270	...	

Table 3. Examples of keywords detected by the method proposed

Fq & N-gram	Fq & N-gram
12 hardware software	19 locations job
12 centre dublin	20 south job description
12 dublin south job description	20 skills ability
12 city centre dublin	20 south job
13 part of team	21 locations city centre job
13 team player	21 software development
13 tech support	21 dublin city centre
13 customer satisfaction	24 dublin city
13 strong knowledge	24 city centre job description
14 work environment	24 project management
16 skills experience	25 years experience
17 excellent communication	25 successful candidate
18 related job description	26 related city
19 locations job description	26 related city centre
19 related job	27 centre job

— The potential to use the key terminology to match job seeker and companies.

— The functionality of using different work sectors in the corpus.

— Use the N-grams in open questions for the companies.

— The volume of real information retrieved.

— The diversity of information content.

— To use the information obtained in the future in conjunction with the CV to make semantic matches.

4 System Overview

We carrying out four different experiments and for all them we used the data preprocessing:

4.1 Data Preprocessing

The following list shows the preprocessing for this research work:

— We explained in section 4 that whole data was downloaded in HTML and JSON files.

— We clean all unnecessary lines such as HTML and JavaScript tags in the corpus.

— The information was stored in different files such as *job1, job2, ... jobn*.

— We created a string with all this information.

— We removed all symbols such as _@, ”, ’, *, ?,_ _, etc._ because the job description is written by the companies and they usually use symbols.

— We convert all the letters in lowercase, because it is the same say *computer science* that *Computer science*, only change the first letter and we had two different bigrams (in this case).

— We used NLTK[4] to tokenize the whole corpus with POS[5] functions because NLTK works by context that is to say use the words before and after of each word, one example is *Support* could be a noun or verb.

— We discard possibles combinations with *”.”*, *”,”* and *”;”*, for example we had a lot of incomplete ideas such as *customers, and providing* and *innovation happens. And*. For this, we developed and use a classification pattern when put a conditional.

For the second experiment, we used a stop words list, to not discard combinations. In Table 3, we can see examples of N-grams used in this research work.

[4] Natural Language Toolkit https://www.nltk.org/

[5] Part of speech

5 Measures and Experiments

We used three types of collocation measures to define the best filter in the N-grams. These measures were chosen for the easy implementation, good results and the low computing power needed with a large volume of information, the following measures have been reported in [11].

— **PMI** Pointwise mutual information is a measure of association:

$$pmi(x;y) \equiv log\frac{p(x,y)}{p(x)p(y)}, \tag{1}$$

$pmi(x;y)$ means the association between two terms (bigram), the first word is represented with x and the second word with y. It is a popular measure for the simple implementation and the good results.

— **Likelihood-ratio** We used *maximum Likelihood-estimation* to decide if there is an important contrast between the expected and the observed frequencies in bigrams, trigrams, and quadrigrams. This measure expected two hypothesis $L(H_1)$ and $L(H_2)$ shown in the formula (2). The following formula describes the occurrence frequency of a bigram w^1w^2:

- **Hypothesis 1.** The occurrence of w^2 is independent of the previous occurrence of w^1:

$$P(w^2|w^1) = p = P(w^2|\neg w^1).$$

- **Hypothesis 2.** It is a formalization of dependence which is good evidence for an interesting collocation:

$$P(w^2|w^1) = p_1 \neq p_2 = P(w^2|\neg w^1).$$

For p, p_1 and p_2 and write c_1, c_2, and c_{12} for the number of occurrences of $w1, w2$ and w^1w^2 in the corpus[11]:

$$log\lambda = log\frac{L(H_1)}{L(H_2)}, \tag{2}$$

$$= log\frac{b(c_{12}, c1, p)b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c1, p_1)b(c_2 - c_{12}, N - c_1, p_2)}, \tag{3}$$

$$= logL(c_{12}, c_1, p) + logL(c_2 - c_{12}, N - c_1, p), \tag{4}$$

$$- logL(c_{12}, c_1, p_1) - logL(c_2 - c_{12}, N - c_1, p_2). \tag{5}$$

We used Chi-square with the same purpose that Likelihood ratio search important contrasts between the frequencies in bigrams, trigrams and quadrigrams, the formula (6) shown how work:

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \tag{6}$$

where i ranges over rows of the table, j ranges over, O_{ij} is the observed value for cell (i, j) and E_{ij} is the expected value.

5.1 Strategy to Compare the Proposed Approach

In this research work, we would like to compare with other research approaches but authors in [12] and [7] do not provide a gold standard to compare with our approach. They provide a corpus, so, the strategy to compare our approach with other approaches is evaluating they corpus and our corpus in the same way.

To achieve this goal is necessary an estimation. One form is using a confidence interval for a population proportion. "A population proportion means the proportion of units in a population that possess some attribute of interest."[6] We used it to estimate the veracity of our results. The formula used it is (7):

$$\pi = P \pm 1.96\sqrt{\frac{P(1-P)}{n}}, \tag{7}$$

where P is the number of your random sample divided by the total number of the sample $\frac{r}{n}$. And the value 1.96 means the 95% confidence interval for a population proportion.

The formula (7) gives us two values. The values mean an approximate confidence interval, in this case with the 95% confidence.

[6] www.statisticalsolutions.ie/wp/content/APPLIED STATISTICS ebook.pdf

Table 4. Sample trigrams filtered by the intersection $A \cap B$

Trigram	Fq	PMI	LKH-R
dublin city centre	53	2019.562	17.231
telecoms tech support	13	501.210	18.167
third level qualification	7	349.176	18.474
benefits competitive salary	3	341.853	18.110
competitive salary earn	2	331.944	21.049
fast paced environment	6	328.250	17.544
equal opportunities employer	6	316.0285	21.500
proven track record	6	306.108	21.363

Table 5. Sample trigrams filtered by the intersection $A \cap C$

Fq & Trigram	PMI	Chi-S
53 dublin city centre	8154393.48	17.23
13 telecoms tech support	3826386.92	18.16
3 successful candidate joining	1258923.95	18.66
3 benefits competitive salary	856803.54	18.11
2 competitive salary earn	4349623.07	21.04
6 fast paced environment	1149592.70	17.54
6 **equal opportunities employer**	17803759.83	21.50

Table 6. Sample trigrams filtered by the intersection $A \cap B \cap C$

Fq & Trigram	PMI	LKH-R	Chi-S
53 **dublin city centre**	2019.56	8154393.48	17.23
13 telecoms tech support	501.21	3826386.92	18.16
3 successful candidate joining	496.63	1258923.95	18.66
7 third level qualification	349.17	2550273.66	18.47
3 benefits competitive salary	341.85	856803.54	18.11
2 competitive salary earn	331.94	4349623.07	21.04
6 fast paced environment	328.25	1149592.70	17.54
6 equal opportunities employer	316.02	17803759.83	21.50

Table 7. Sample trigrams filtered by the intersection $B \cap C$

Trigram	Fq	LKH-R	Chi-S
related locations dublin	61	2371.108	1662990.691
centre job description	24	2202.153	994671.932
south job description	20	2198.519	1503450.729
cork job description	14	2067.289	539817.166
limerick job description	9	2021.597	559561.417
dublin city centre	53	2019.562	8154393.488
waterford job description	6	1987.107	501852.430
locations dublin city	31	1787.154	1764364.726

5.2 Intersection

We implemented Likelihood-ratio positive because we are only interested in positive results. A positive result means an evaluation of the occurrence of an N-gram in the corpus and a negative result is the evaluation that an N-gram does not occur in the corpus. We create a filter derived from the aforementioned measurements, we take the results of each one and we intersect them giving a subset. That is to say, each one has its own range, so only took the best results of each one. We represent the set PMI as set A, Likelihood-radio as set B and Chi-square as set C. Thus we get the following intersections.

In Table 4, we can observe $A \cap B$ (see Fig. 1). This intersection between two sets of values PMI and Likelihood-ratio, where both have high values and we see the 10 first trigrams with the highest value. To do the intersection only 50% was taken that is to say one subset from A and another B. You can see the difference in the N-gram *competitive salary earn* has in PMI 331.944 higher than Likelihood-ratio with 21.049.

In Table 5, we can observe $A \cap C$ (see Fig. 1). This intersection between two measures PMI and Chi-square, where both have high values and we see the 10 first trigrams with the highest value. In special the term *equal opportunities employer* start to obtain key terminology. If compared Table 4 with Table 5 some trigrams are removed.

In Table 7, we can observe $B \cap C$ (see Fig. 1). This intersection between two measures Likelihood-ratio and Chi-square, where both have highest values and we see the 10 first trigrams with the highest value. We can see that measure Chi-square discard N-grams because their N-grams have low values and when we used different percentages, in view of Likelihood-radio gave us N-grams with higher values, these values were discarded in this intersection.

In Table 6, we can observe $A \cap B \cap C$ (see Fig. 1). This intersection between three measures PMI, Likelihood-ratio and Chi-square. It is one of the main objectives of this research work, because we can observe how begin to filter the information. You can see the respective measure of each one. When comparing the Tables 4, 5 and 7, we can see that Chi-square measure discarded more N-grams.

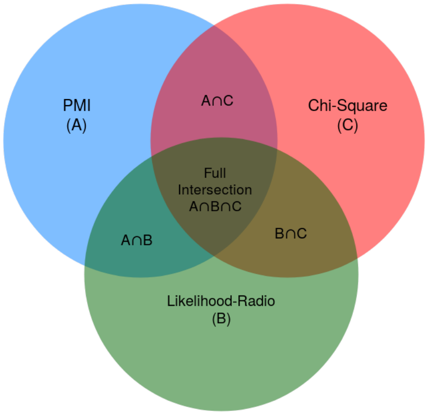


Fig. 1. Set intersection

Table 8. Sample of trigrams filtered by the intersection process

Fq & Trigram	LKH-R	Chi-S	PMI
61 related locations dublin	2371.1	1662990.69	14.73
24 centre job description	2202.15	994671.93	15.31
20 south job description	2198.51	1503450.72	16.17
14 cork job description	2067.28	539817.16	15.17
9 limerick job description	2021.59	559561.41	15.85
53 dublin city centre	2019.56	8154393.48	17.23
6 waterford job description	1987.1	501852.43	16.27
3 job description summary	1943.12	295844.58	16.44

Table 9. Trigram with set intersection and filter with POS

Fq & Trigram	LKH-R	Chi-S	PMI
61 related/JJ locations /NNS dublin/NN	2371.1	1662990.69	14.73
20 south/NN job/NN description/NN	2198.51	1503450.72	16.17
14 cork/NN job/NN description/NN	2067.28	539817.16	15.17
9 limerick/NN job/NN description/NN	2021.597	559561.41	15.85
53 dublin/NN city/NN centre/NN	2019.56	8154393.48	17.23

5.3 Experiment 1

Experiment 1 presents the set intersection. The set intersection is when the results of each measure match in the ranking and does not comply with the POS patterns for one experiment and does comply with the POS patterns for the rest. This set intersection is between the PMI, Likelihood-ratio and Chi-square measures used to rank terms but without POS filter and order by Likelihood-ratio. We can see the different results in the Table 8 and 9.

In Table 8, we can observe that the measures Chi-square and PMI are not congruent in a descending or ascending form. This is due to the fact that many terms were discarded by the intersection. The Likelihood-radio results are ordered in a descending form but between each value, there is a big difference, this is also due to the fact that N-grams were discarded.

To explain better why N-grams are discarded when the intersection of the three measurements is done. It is necessary to know that an intersection is a subset of other sets, in this case of three sets (measures). We call full intersection to this subset (see Fig. 1).

5.4 Experiment 2

Experiment 2 is defined by the intersection of sets generated by the three collocation measures defined and a POS filter. We also used tokenization with POS tags. The POS filter consists in to verify if the first word is labeled by a *JJ* or *NN* followed by any other tag or couple of tags and ending with a tag *NNS* or *NN*. For instance, in Table 9 we can see N-grams filtered by discarding mainly verbs.

In Table 10, we can observe the N-grams that did not follow the POS pattern defined. We can see a pattern at the beginning of the N-grams that start with the following tags: *IN*, *VB*, *VBG* or *RB*. Taking into account this pattern, the filter was created discarding all the N-grams that had that pattern. We called this discarding as POS filter.

It is important to note that we only defined the POS pattern at the beginning and at the end of the N-grams that means that in the middle of the N-grams could be any other N-grams with any POS tag.

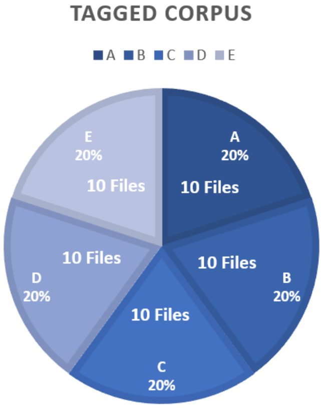


Fig. 2. Labeled corpus

Table 10. Trigram with set intersection and tokenized but without filter POS

Fq & Trigram	LKH-R	Chi-S	PMI
2 ensure/VB customer/NN satisfaction/NN	223.51	41012.61	14.24
2 across/IN multiple/NN projects/NNS	208.22	69104.1	15.03
2 establish/VB best/JJS practice/NN	177.99	3266621.42	20.63
2 across/IN multiple/NN time/NN	176.72	92012.22	15.45
3 rewarding/VBG work/NN environment/NN	170.57	191825.61	15.96

Table 11. Manual tags results

Freq.&Patterns	Freq.&Patterns	Freq.&Patterns
1000 NN NN NN	97 JJ NNS NN	48 NN NNS VBG
560 NN CC NN	94 VBG NN NN	48 JJ JJ NN
551 NN IN NN	90 NN NN VBG	45 NNS IN NNS
422 NN NN NNS	90 NN IN VBG	44 NN CD NN
303 JJ NN NN	87 NN VBG NN	44 JJ IN NN
244 NN NNS NN	71 NNS JJ NN	41 NNS VBG NN
242 NN TO NN	67 VBG IN NN	40 NN PRP$ NN
196 NNS NN NN	67 VBG DT NN	39 NNS NNS NN
195 NN JJ NN	66 NN CC NNS	39 CD NN NN
193 NNS IN NN	64 NNS NN NNS	38 NN VBN NN
180 NN DT NN	57 VB DT NN	38 NNS JJ NNS
163 NNS CC NNS	56 VBG NN NNS	37 JJ JJ NNS
157 NN IN NNS	56 NN NNS NNS	36 NN NN VBN
152 NNS CC NN	55 NN VBG NNS	35 NN VBZ VBN
142 NN JJ NNS	53 VBG CC VBG	35 NNS IN VBG
135 JJ NN NNS	52 NN CC VBG	35 NNS DT NN
104 NNS TO NN	49 VBG JJ NN	35 JJ CC NN

5.5 Experiment 3

In the manual labeled corpus also called *positive tags*. The reason for the name *positive tags* is because one human expert labeled careful each keyword. The size of the labeled corpus is 50 job descriptions. For each job description file, there is another file with labels contained bigrams, trigrams, and quadrigrams with the next structure:

— BIGRAM: word1 word2,

— TRIGRAM: word1 word2 word3,

— QUADRIGRAM: word1 word2 word3 word4.

We discovered four hundred twenty-four patterns of which one hundred fourteen only have frequency 1, Seventy-three have frequency 2, thirty-four have frequency 3, twenty-three have frequency 4, nineteen have frequency 5, sixteen have frequency 6, twelve have frequency 7, nine have frequency 8 and nine have frequency nine. With previous numbers, we decided to remove patterns with frequency from 1 to 9.

Since if we take a range higher such as frequency from 1 to 15 the recall measure rapid decreases because above of frequency 9 there are many keywords that depend on it. In an opposite way, when is below of frequency 9 the recall measure rapid increase but the precision measure decrease and at the same time F-measure decrease.

In total were three hundred nine patterns removed. In addition, the proposed approach will have problems with the recall measure because eight hundred eighty-two tags do not exist in the results.

The main patterns results are in the Table 11 they frequency is between thirty-five and one thousand and they were the patterns that gave us good results for the next experiment. We can see the patterns diversity than we occupy in this research work to develop the automatic key extraction. In the first row appears one thousand times that is mean that is very important. Also, we can see that a lot of them start with *NN, NNS, VBG* or *JJ*. Also, the first three in the Table 11 gave us unnecessary keywords, for this case is like a balance the POS pattern gave us necessary and unnecessary keywords.

Once we got the results we decided to create the POS patterns filter only taking the beginning and the end (see Fig. 3). We discovered that filter comes more unnecessary tags in the final results because the precision it was lowest.

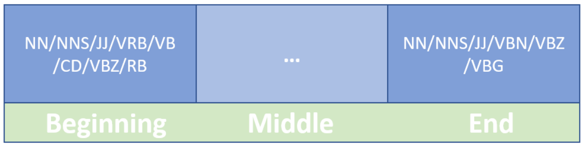
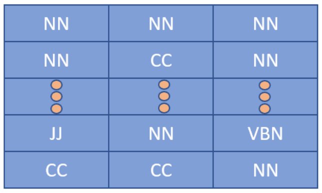


Fig. 3. First POS filter



NN	NN	NN
NN	CC	NN
⋮	⋮	⋮
JJ	NN	VBN
CC	CC	NN

Fig. 4. Second POS filter

Then we decided to use exactly the POS patterns extracted from the manually labeled corpus to filter and reduce the unnecessary keywords. This action had an increase in precision measure and F-measure.

We can observe five groups (see Fig. 2) and each group contain ten job description manuals labeled. With this corpus, we can compare the results obtained. It was training with three parts of the corpus and then take the POS to prove in the rest. There are ten possible combinations:

Set for POS pattern extraction	Testing set
(A + B + C)	(D + E)
(A + B + D)	(C + E)
(A + B + E)	(C + D)
(A + C + D)	(B + E)
(A + C + E)	(B + D)
(A + D + E)	(B + C)
(B + C + D)	(A + E)
(B + C + E)	(A + D)
(B + D + E)	(A + C)
(C + D + E)	(A + B)

Where the first column is the set used to train the proposed approach with that patterns and the second column is the set used to prove it.

For this comparison only need the *positives tags* but for the second we needed the *negatives tags*. The *negative tags* were obtained from the ones that are not in the labeled corpus set.

When we add the negatives we create a new set (see Fig. 5). The new set was bigger than the first one, now contain 100 jobs descriptions divided into 5 groups, each group contain ten *positive tags* and ten *negatives tags*. There are 10 possible combinations:

Set for POS pattern extraction	Testing set
(A + B + C + AN + BN + CN)	(D+E+DN+EN)
(A + B + D + AN + BN + DN)	(C+E+CN+EN)
(A + B + E + AN + BN + EN)	(C+D+CN+DN)
(A + C + D + AN + CN + DN)	(B+E+BN+EN)
(A + C + E + AN + CN + EN)	(B+D+BN+DN)
(A + D + E + AN + DN + EN)	(B+C+BN+CN)
(B + C + D + BN + CN + DN)	(A+E+AN+EN)
(B + C + E + BN + CN + EN)	(A+D+AN+DN)
(B + D + E + BN + DN + EN)	(A+C+AN+CN)
(C + D + E + CN + DN + EN)	(A+B+AN+BN)

Where the first column is the set used to train the proposed approach with patterns combined positives and negatives and the second column is the set used to prove it also combined with positives and negatives.

5.6 Experiment 4

In this experiment, we decided to use three measures to increase the precision and F-measure. To obtain better results we decided to use an intersection between the measures. There are for possible combinations:

— $A \cap B$: PMI and Likelihood-Radio see Fig. 7.

— $A \cap C$: PMI and Chi-square see Fig. 6.

— $B \cap C$: Likelihood-Radio and Chi-square see Fig. 10.

— $A \cap B \cap C$: PMI, Likelihood-Radio and Chi-square see Fig. 8.

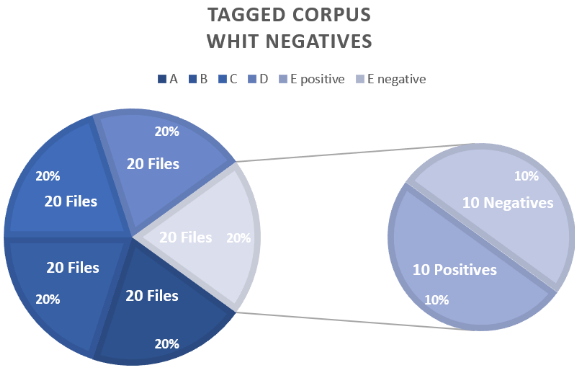


Fig. 5. Labeled corpus with negatives

As each measure has its own value, we decided that we would have to base ourselves on different percentages in order to be able to discard any possible unnecessary labels.

From the intersection results, the values were sorted and some percentages were extracted. We decided to get the following percentages: 30%, 50%, 60%, 70%, 80%, 90% and 100% in order to determine the best subset with the highest number of keywords. For this paper we refer to the percentages as sets, with the following names, 30% PA, 50% PB, 60% PC, 70% PD, 80% PE, 90% PF, and 100% PG. The reason was to avoid having a single focus and prove which one gives us better results.

Each intersection has the its own results which are explained with supporting detail in the the following graphics.

In Fig. 6 we can see the intersection between PMI and Likelihood-Radio. The recall increases faster in each percentage, but we can see when PA starts, it is the lowest but it becomes the highest measure.

The precision measure in PA is higher because there are not many keywords and the few keywords are in the set labeling manually. As it was expected after the PA was the lowest with a slight decrease in each one because the number of keywords has rapid increase but the keywords are not in the set labeling manually. The F-measure start between recall and precision measure and after that always had a slight increase.

In this intersection we can recognize the higher value for each one, for recall of PG was 0.6896, because when more files are used the probability to find a keyword that matches with the corpus labeled manually increases. The precision measure of PB was 0.1326, this happens because it was a balance in the numbers of keywords obtained for the proposed approach matching with the corpus labeling manually. Finally, the F-measure was PG with 0.2139, this happens because the F-measure is the combination between two measures: recall and precision, so, when the recall measure has a high value this implies the F-measure will increase.

In Fig. 6 we can see the intersection between PMI and Chi-square. we can see the same behavior that graph (see Fig. 7) the recall increase faster in each percentage first start with the lowest percentage but after that always remain in the head and his higher value is PG with 0.6896, this happen for the same reasons when the corpus is big, so there are more keywords to math with the corpus labeling manually. For precision had a slight decrease and his higher value is 0.1454. Finally for F-measure is PG with 0.2139, although is a different intersection using different measures throw almost the same results.

In Fig. 10, we can see the intersection between Likelihood-Radio and Chi-square. We observe this intersection has the same behavior that the previous graphs (see Fig. 6 and 7), but change the data results. For this intersection is interpreted as the previous ones. Showing the importance of the POS filter.

In Fig. 8, we can see the intersection between PMI, Likelihood-Radio and Chi-square. The higher value for recall is PG with 0.6896 and the lowest value is PA with 0.0922.

For the precision measure with the higher values is PA with 0.1454 and the lowest is PE with 0.1245.

Finally, F-measure with the higher value is PG with 0.2139. The comparison of the graphs (see Fig. 7, Fig. 6 and Fig. 10) have the same behavior and this can be explained because the POS patterns are useful to reduce the unnecessary keywords extracted by the collocation measures. Although, it should be noted that have the same behavior but no the same values in the results,

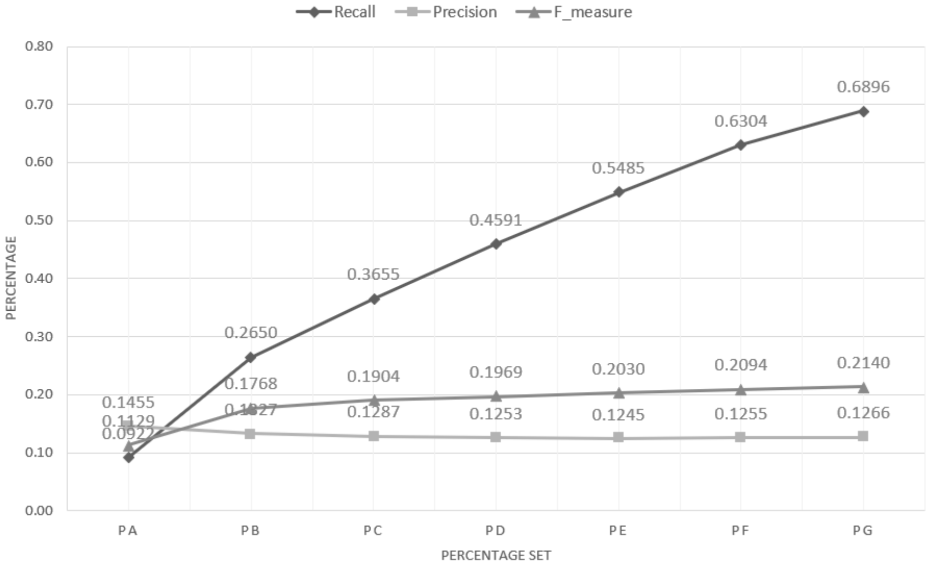


Fig. 6. Intersection $A \cap C$

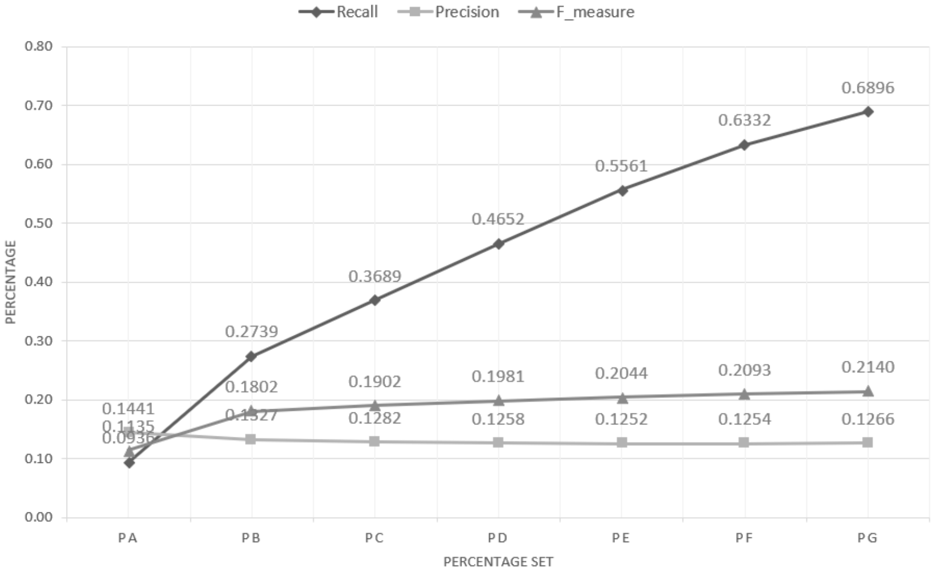


Fig. 7. Intersection $A \cap B$

and we can say that for this research work the best experiment was the intersection using the three measures.

It should be mentioned that the PG of each intersection is exactly the same, this is because when using the whole information does not have any label discards. In the graph (see Fig. 9) we observe the results between the PG set with and without POS patterns filter. The first measure to explain is recall, in this one, we observe that the

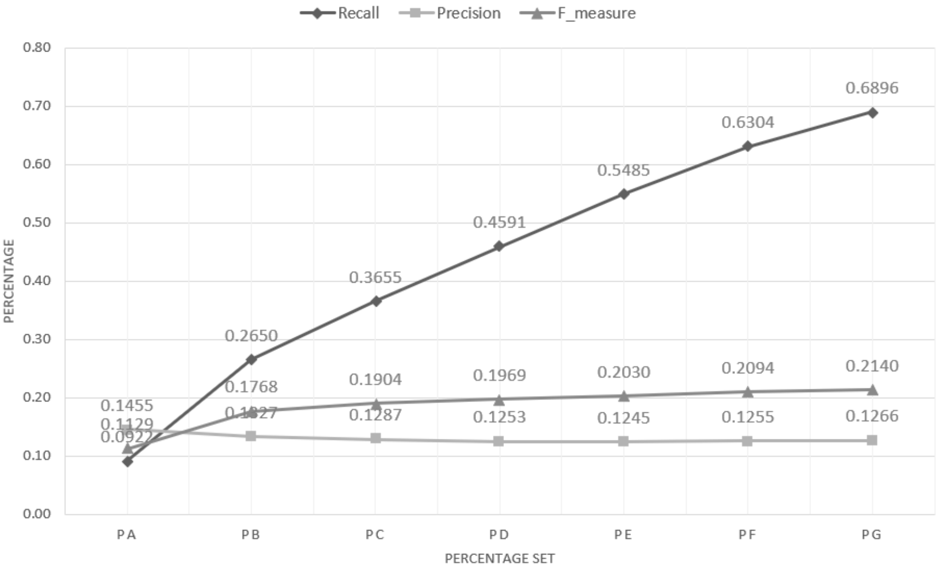


Fig. 8. Intersection $A \cap B \cap C$

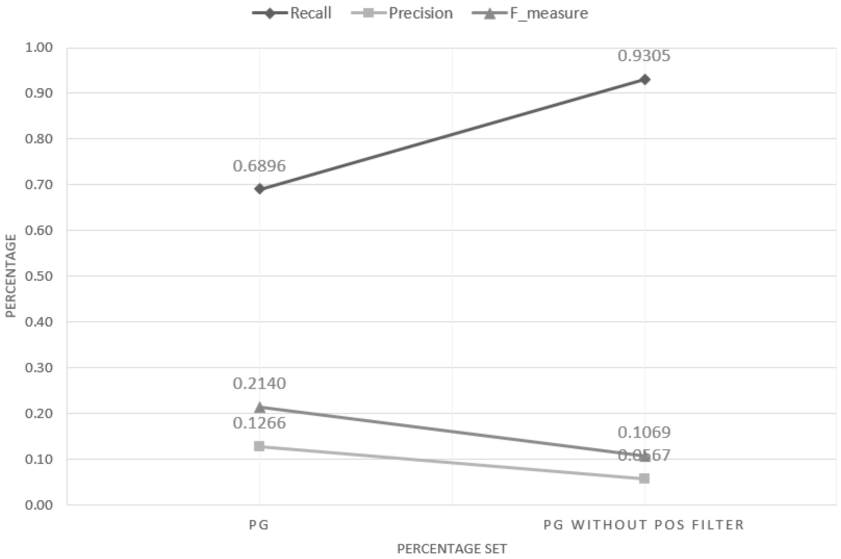


Fig. 9. Comparison of results

difference is 0.2408 being the higher that does not has POS filter. This easy to interpret, because when it does not have the POS patterns filter the proposed approach extracted a lot of keywords gave us a higher value for recall measure. In the same way when there are a lot of keywords that are not in the corpus labeling manually reduce the precision measure and increase when has the POS filter the difference is 0.0699. Finally for F-measure is a combination between precision

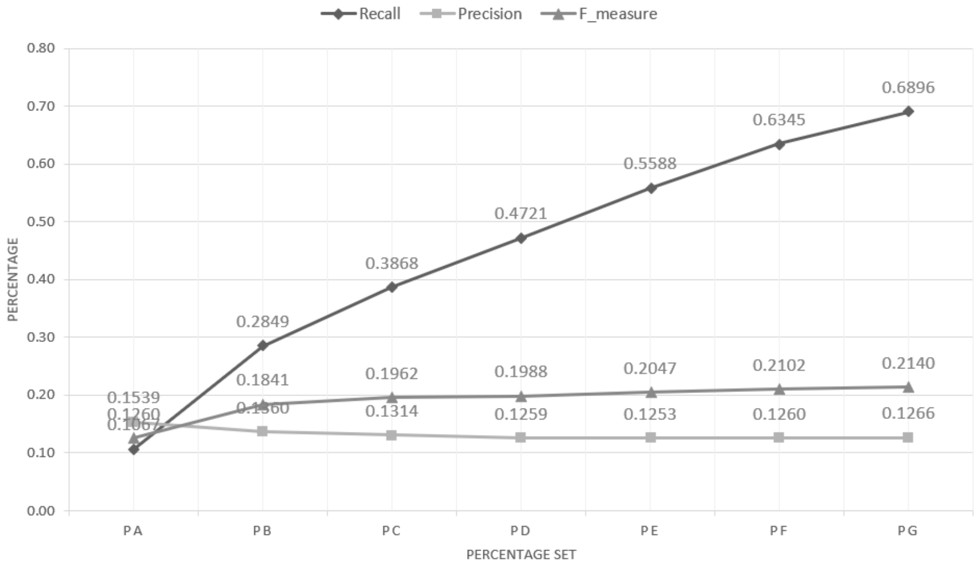


Fig. 10. Intersection $B \cap C$

measure and recall measure so for this research work is a high importance. We can see that the higher is the first one with 0.2139 and the difference is 0.107, thus we can say that the POS patterns filter method is better.

In summary, the last pattern lost 24% in recall measure but increase in precision with 6.9% and the F-measure with 21.3%.

5.7 Comparison with other Dataset

In this experiment, we used another corpus to demonstrate that POS patterns work not only in job descriptions.

The information has 454,580 N-grams divided in two:

- *Good*: they comply with our filter pattern.
- *Bad*: they do not comply with our filter pattern.

In Fig. 11, we observe the PMI behavior. The gray line means the *Good* and the black line means the *Bad*. The *Good* means the N-gram is correct because comply with our patterns and the *Bad* means the appositive. In this pattern, we observed that PMI is good with a large information volume. The *Good* part always remains above the *Bad* with a big difference.

In Fig. 12, we observe the Likelihood-ratio behavior. In this behavior begins with the same proportion of *Good* and *Bad*. But by 100,000 the percentage of *Bad* is a little greater than *Good*. Then it changes a little before reaching 400,000 for good ones have a slightly higher value. This happens because the Likelihood-radio put a value to each trigram and that value put high values to trigrams that no have the POS pattern.

In Fig. 13, we observe the Chi-square behavior. In this behavior we can see that it looks like PMI behavior. With the difference that a little less distance between the *Good* and the *Bad*. Remaining with a higher value the *Good* ones. This happens because the Chi-square put similar values to trigrams that do comply with POS patterns and those that do not comply as well.

In Fig. 14, we observe the subset *full intersection* without sorted. Here we can see the three collocation measures together and the *Good* percentage. Likelihood-ratio stays below the other two measures. In the beginning, it was not constant. We can see that the one maintained

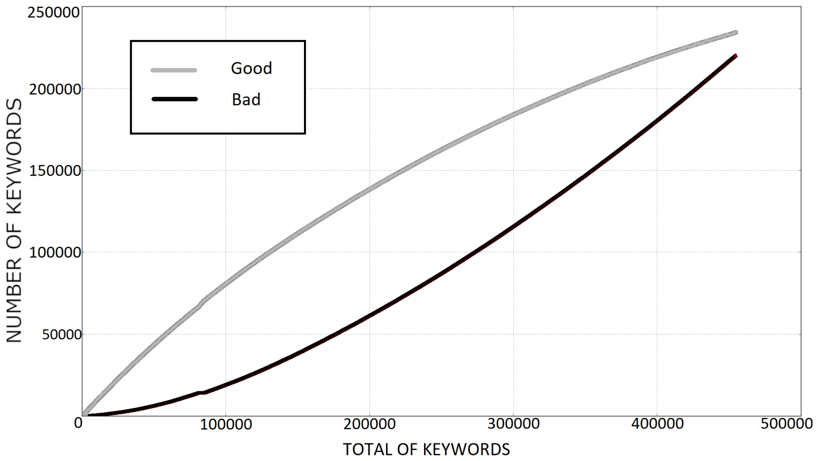


Fig. 11. PMI measure

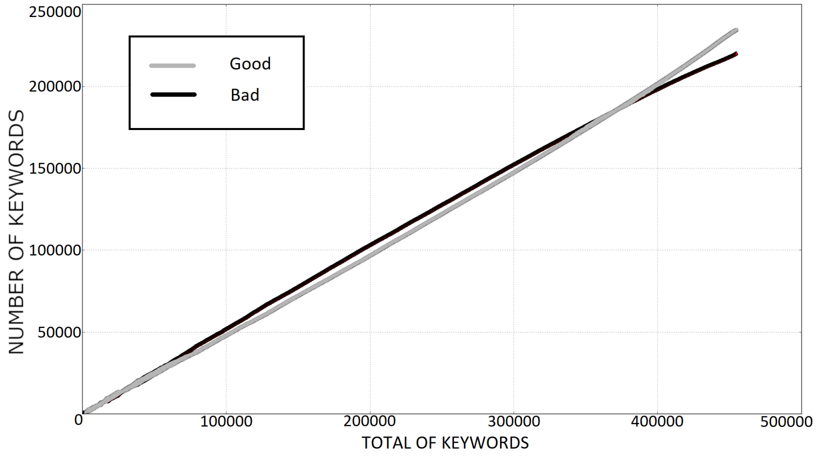


Fig. 12. Likelihood-ratio measure

with better results was PMI but is very similar to Chi-square.

In Fig. 15, we observe the subset *full intersection* with sorted. In this graph unlike the Fig. 14 are ordered seeing three important things:

— Likelihood-ratio remains stable for almost the entire corpus. How it is explained in the *section 5 Measures (Likelihood-ratio)*, this happens because *the occurrence of w^2 is independent of the previous occurrence of w^1*. Thus, it remained consistent in its behavior.

— Chi-square is the second best for this research work. How it is explained in the *section 5 Measures (Chi-square)*, this happens because Chi-square searches important contrast between the frequencies. Thus, it depended on the corpus size.

— PMI, In this case, we observe this measure obtain better results than the other two. How it is explained in the *section 5 Measures (PMI)*, PMI is the probability of a particular co-occurrence of events $p(x, y)$. Thus, it obtained the higher values for each collocation.

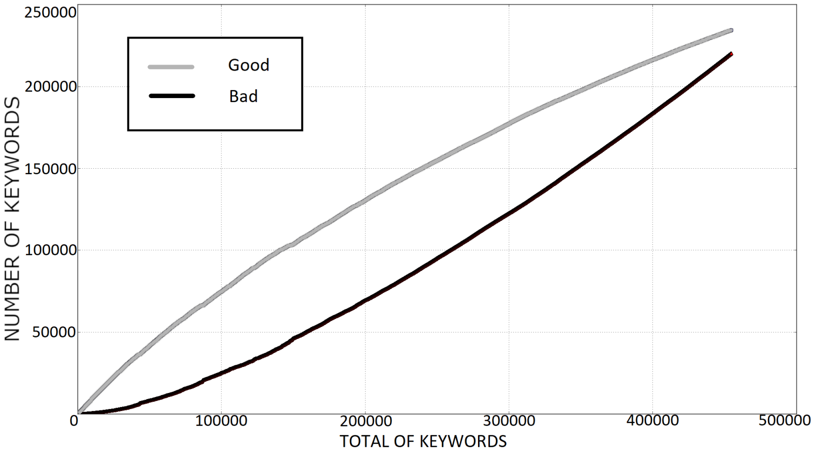


Fig. 13. Chi-square measure

6 Verifying the Results

For this research work, we wanted to have a point of refining to verify the results. To achieve this objective we take a 10% random sample of N-grams. Because the number of labels in 10% gives us an margin error of 3.8%. The objective for this research work is not to label a large volume of documents, thus, the margin error of 3.8% is enough. Each N-gram was labeled with an ”y” if the label has a valid keyword, and an ”n” label if it was not a valid keyword. In order to verify and compare our results with respectable accuracy. We implemented the formula (7) described in section (5.1). Resulting in two intervals.

The first interval is when we could get valid keywords that are labeled with an ”y”:

$$0.9134 < \pi < 0.9597. \qquad (8)$$

The interval (8) means that the possibility to find a valid keyword is between 91.34% and 95.97%. We consider this interval as a significant value of accuracy, for this research work.

The second interval is when we could get an invalid keyword that are labeled with ”n”:

$$0.1615 < \pi < 0.2374. \qquad (9)$$

In an opposite way, we show the possibility to get invalid keywords in the interval (9) between 16.15% and the 23.74%. That is a reasonable interval of ”n” labels if it is compared with the interval of ”y” labels (8).

Comparison with another corpus made of articles used in [12] and evaluated in the same way, taking a random sample with a margin error of 3.8%. Also, the sample was labeled manually with ”y” or ”n”, even that the corpus is made of articles of different categories.

When we were labeling the sample, we had a few errors in the beginning because contains names in Urdu and Chinese. Likewise, the task of labeling was difficult because we were not familiarised with the content of each article. So, to compare the result is using the same formula (7).

The first interval is using the ”y” labels:

$$0.8351 < \pi < 0.8865. \qquad (10)$$

We can see in the interval (10) that has a percentage between 83.51% and 88.65%. Although, it is not as accurate as the interval (8) is a high level of percentage of valid keywords:

$$0.1095 < \pi < 0.1602. \qquad (11)$$

In the interval (11) we can see that the interval of invalid keywords is lower than the interval (9). That is a favourable signal that the patter proposed is convenient to obtain keywords for other corpora, because if compare the ”y” labels in the interval (10) and (8) both of them have high values to obtain a valid keyword.

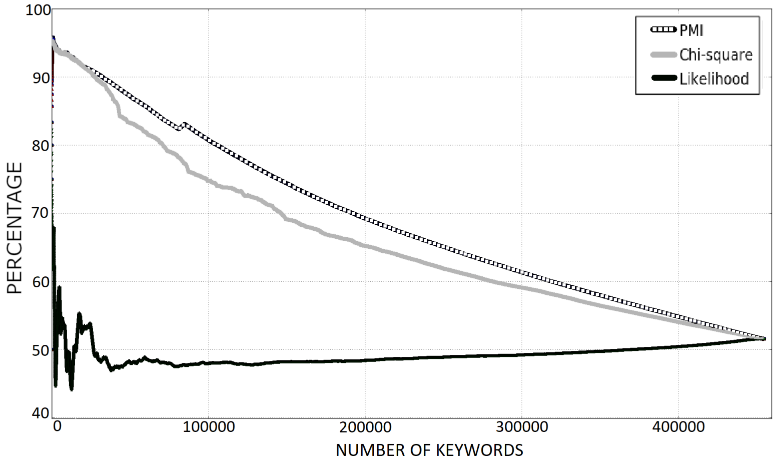


Fig. 14. Measures intersection without order

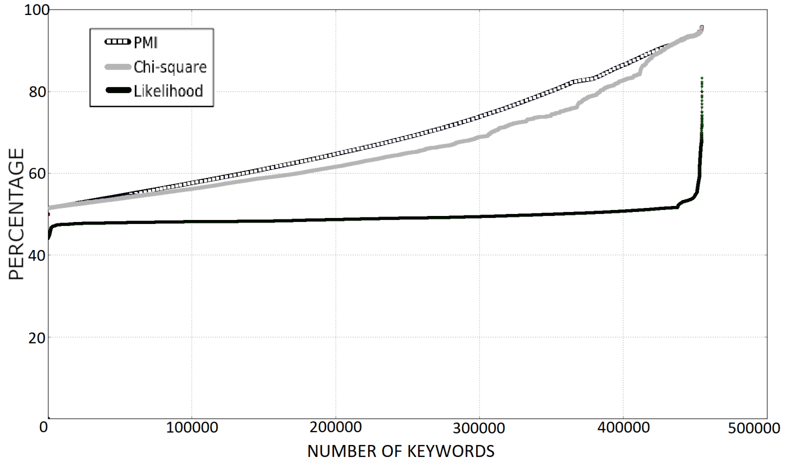


Fig. 15. Measures intersection with order

7 Conclusion

We started labeling 50 Jobs description files with N-grams (bigrams, trigrams, and quadrigrams). These were called manual labels, which were fundamental to later compare them with the labels that were obtained by the proposed approach. When doing the labeling we decided to achieve their respective POS patterns, four hundred twenty-four patterns were discovered. Of which we decided to count how many times that POS patterns were repeated within all the labels. Therefore, they were ordered by frequency.

It is important to say that the pattern that has the most frequency (*NN NN NN*) is also one that does throw labels that are not in the manually labeled corpus making the precision measure low and therefore the F-measure too.

In experiment 1 and 2, we proceeded to give a weight to each label for this we use measures such as PMI, Chi-square, and Likelihood-ratio, but for this, we wanted to make another type of filter for the labels.

So, an intersection was created between them to see if this improved, in this experiment we observed that the four intercessions have a similar behavior. What varies are the values in the results we can see that some have more or less value. This can be explained of the four intersections is that the intersection that has the three measures is what gives us better results.

In experiment 3, we present two proposals for using POS patterns to use the POS patterns obtained, one was using the first and last word of the N-grams. In Conclusion of this proposal is that it is not very good since it generated many of the patterns that did not match the manually labeled patterns. So, we had the second proposal of exactly using the POS patterns, but the precision was still very low, consequently in the last idea was slightly modified, removing the patterns that had frequency between 1 - 9 and that increased the precision and F-measure, but a slight decrease in recall measure the results were presented in the previous graphs.

It should be noted in experiment 4 that the graph (see Fig. 9) shows the comparison between the intersection of the measurements with and without the POS filter. We can see that the recall decreases 24% but here we can also discuss that it decreased since we left out the patterns that had frequency 1 - 9 that also influenced that part. We can also see that it had an increase in the precision measure (6.9%) and F-measure (10.7%).

We also wanted to see if what was implemented in this research work applied to other corpora, so we proceeded to the implementation in the yelp[7] corpus of reviews. In these results, good labels are observed and meaningful so it can be said that these patterns can be applied in other corpora. In the same way in section 6, we implemented a method to verify the accuracy of our results and the comparison with another corpus, concluding that these experiments can be applied in a different corpus and obtain a high percentage of valid keywords.

A future work would be to use the POS pattern with different methods such as TF-IDF[6], TextRank[13], and RAKE[16] because they are the top extractors of keywords. Also, there is the possibility to improve the results if the methods are combined with the proposed POS patterns obtained in this research work. Another task would be to use these obtained terms to feed automatic learning algorithms such as embedding words and convolutional neural network.

[7] https://www.yelp.com/dataset/challenge

References

1. **Bharti, S.K., Babu, K.S., & Jena, S.K. (2017).** *Automatic keyword extraction for text summarization: A survey.*
2. **Bird, S. (2006).** NLTK: The natural language toolkit. *Proceedings of the COLING/ACL on interactive presentation sessions*, pp. 69–72. DOI: 10.3115/1225403.1225421.
3. **Brezina, V., McEnery, T., & Wattam, S. (2015).** Collocations in context. *International Journal of Corpus Linguistics,* Vol. 20, No. 2, pp. 139–173. DOI: 10.1075/ijcl.20.2.01bre.
4. **Habibi, M. & Popescu-Belis, A. (2015).** Keyword extraction and clustering for document recommendation in conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* Vol. 23, No. 4, pp. 746–759. DOI: 10.1109/TASLP.2015.2405482.
5. **Jurafsky, D., & Martin, J.H. (2014).** *Speech and language processing.* Pearson, Vol. 3.
6. **Lee, S. & Kim, H. J. (2008).** News keyword extraction for topic tracking. *Networked Computing and Advanced Information Management. Fourth International Conference on,* Vol. 2, pp. 554–559. DOI: 10.1109/NCM.2008.199.
7. **Le, T.T.N., Nguyen, L.M., & Shimazu, A. (2016).** Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. *Advances in Artificial Intelligence: 29th Australasian Joint Conference*, pp. 665–671. DOI: 10.1007/978-3-319-50127-7_58.
8. **Liu, F., Pennell, D., Liu, F. & Liu, Y. (2009).** Unsupervised approaches for automatic keyword extraction using meeting transcripts. *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics,* pp. 620–628.
9. **Luthra, S., Arora, D., Mittal, K., & Chhabra, A. (2017).** A statistical approach of keyword extraction for efficient retrieval. *International Journal of Computer Applications,* Vol. 168, No. 7, pp. 31–36. DOI: 10.5120/ijca2017914443.

10. **Maldonado-Guerra, A. & Emms, M. (2011).** Measuring the compositionality of collocations via word co-occurrence vectors: Shared task system description. *Proceedings of the Workshop on Distributional Semantics and Compositionality,* pp. 48–53.
11. **Manning, C. D. & Schutze, H. (1999).** *Foundations of statistical natural language processing.* MIT press.
12. **Martinez-Romo, J., Araujo, L., & Duque-Fernandez, A. (2016).** SemGraph: Extracting keyphrases following a novel semantic graph-based approach. *Journal of the Association for Information Science and Technology,* Vol. 1, pp. 71–82. DOI: 10.1002/asi.23365.
13. **Mihalcea, R. & Tarau, P. (2004).** Textrank: Bringing order into text. *Proceedings of the conference on empirical methods in natural language processing,* pp. 404–411.
14. **Mondal, A.K., Maji, D.K., & Karnick, H. (2004).** Improved algorithms for keyword extraction and headline generation from unstructured text. *First Journal publication from SIMPLE groups, CLEAR Journal*.
15. **Oxford dictionaries (2018).** http://www.dictionary.com/browse/state-of-the-art.
16. **Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010).** Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory,* pp. 1–20. DOI: 10.1002/9780470689646.ch1.
17. **Shi, W., Zheng, W., Yu, J.X., Cheng, H., & Zou, L. (2017).** Keyphrase extraction using knowledge graphs. *Data Science and Engineering,* Vol. 2, pp. 275–288.
18. **Slobodan B. (2014).** *Keyword extraction: A review of methods and approaches.*
19. **Teneva, N. & Cheng, W. (2017).** Salience rank: Efficient keyphrase extraction with topic modeling. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics,* Vol. 2, pp. 530–535. DOI: 10.18653/v1/P17-2084.

Article received on 30/10/2019; accepted on 09/03/2020.
Corresponding author is David Pinto.