Comparison of string metrics effectiveness for the purpose of estimating the number of unique job offers

Marek Zachara, Cezary Piskor-Ignatowicz

AGH University of Science and Technology, Department of Automatics

Abstract: The article presents the results of search for a text-comparison method applicable for identifying same or similar job offers. This is done by calculating pairwise similarity metrics between offers using well known metrics (i.e. Levenshtein, Jaro-Winkler and Jaccard). The article assesses the effectiveness of the algorithms and their applicability to the task. Issues related to processing of data off the web pages and computational requirements are also discussed.

Keywords: string metrics, text matching, offers estimation, automated web pages processing

1. Introduction

The following article discusses various aspects of automated text comparison methods applied to the data retrieved from the Internet. All experiments are centered around a real world example of comparing job offers available on several major Polish job sites (*infopraca.pl*, *jobs.pl*, *praca.pl*, *etc.*). The research was performed on a request from an organization that monitors the state of local job markets. Although the request was quite specific, the conclusions from this research can be generalized, since the methods and algorithms utilized were not manually tuned for the specific task.

All of the above mentioned job services offer flexible search criteria that allow for selecting offers supplied within specified time frame and narrowed down to certain region. Therefore, since the goal was to identify the number of unique job offers posted every day by employees, the primary concern was to eliminate duplicate offers from the considered pool. By duplicates we consider both the same offer posted on several job sites, as well as the same offer re-appearing on the same site within certain period of time (e.g. two weeks). Ability and certainty of identifying such duplicates are discussed later in this article.

2. Text comparison methods

Even though it's easy to verify if two texts are identical or not, there is no obvious and absolute method that would measure a similarity between them. Since compared offers may differ slightly (e.g. in wording) and yet represent the same offer, a similarity metrics is needed that would express the difference between the texts, preferably in form of a number proportional to their likelihood, thus allowing for robust classification of duplicates.

Comparing arbitrary character sequences is an area that has been given attention for a long time. It is constantly put to new uses, e.g. detecting plagiarism [4] or monitoring competition [3].

A very good account of currently available algorithms and their performance is given in [7]. This publication also categorizes the available methods into two major groups: *substring search* methods and calculation of *edit distance*. Since both compared texts are usually similar in length and form, the later group, which focuses on calculating a 'cost' of transforming one text into another better suited for the task. From this group, several well-known and commonly used methods have been taken into considerations, namely:

- Levenshtein distance [2],
- Jaro-Winkler distance [8],
- Jaccard similarity coefficient [1].

It is worth noting, that especially the latest one is not bound to character-level comparison, but instead can work on words or n-grams, which are groups of characters of words [5]. This is taken into account when performing the experiments.

3. Processing of web sites

The referenced text-comparison methods have been developed for processing of what can be called a 'plain text', i.e. text that does not include any formatting or mark-up. In the described scenario however, the data is available as a set of web pages, which form the respective web site.

This adds an additional layer of difficulty to the task, as interesting information is interleaved with the HTML markup and the layout of the web site. The earlier (HTML markup) is easy to deal with as the tags can be relatively easily removed from the retrieved page. The additional content is however much more difficult to remove – and leaving it would seriously impair the quality of duplicates' classification. To deal with this problem, relative frequency of occurrence of each phrase is calculated across each web site. With large enough sample it is possible to identify the parts of the pages pertaining to the web site itself with a high certainty – as it is shown later in the article.

4. Experiment arrangement

For the purpose of experiments, a base set of over 10 000 job offers had been retrieved from 5 major polish job portals

(1500–3200 from each site). For the efficiency reasons, some of the experiments were run on a randomly selected subset of this set (if so, specific samples are mentioned together with the results). It is important to point out, that the number of comparisons needed equals to 2-combination of the selected set, therefore computing time rises greatly with larger sets.

Also, the computational complexity of the text-comparison algorithms strongly depends on the length of the compared texts. One of the most efficient algorithm [6], recommended in [7] has computational complexity of $n*m/\log n$. Unfortunately, job offers has length in range of 1000 bytes or more, which contributes heavily to the computational workload.

5. Results

As it was mentioned in the first paragraph, the web pages retrieved from the Internet needed to be cleaned up of all the irrelevant data. As can be seen in tab. 1, the amount of irrelevant data outnumber the valuable data by the ratio of 10–20 (average job offer is around 1500 bytes). 'Markup' represents all the HTML tags, scripts, etc. While 'Common phrases' are phrases repeatedly found in at least 50 % of all the web pages retrieved from the web site.

Tab. 1. Average pages size and their composition. Values in bytes. Numbers in parenthesis are standard deviation

Tab. 1. Przeciętne wielkości stron i ich kompozycja. Wartości podane w bajtach. W nawiasach podano odchylenie standardowe

	Retrieved page size	Markup	Common phrases
job site 1	39 272 (4503)	36 147 (3715)	1 849 (161)
job site 2	26 507 (1112)	23 539 (650)	1 497 (13)
job site 3	22 093 (1700)	16 631 (1192)	1 847 (110)
job site 4	30 719 (2654)	27 142 (2380)	1 312 (36)
job site 5	17 890 (994)	13 985 (582)	2 091 (6)

Failing to reliably remove even the common content would seriously impair the effectiveness of the comparison as it tends to be similar in size to the job offer itself.

Cleaned up job offers were then compared against each other and their similarities were measured with three mentioned metrics. In the fig. 1–3 the applicable results are presented. The horizontal line in each of the figures denote assessed similarity threshold level, explained in more detail below.

As it turns out, it is hard to reliably define the similarity level that would distinguish ‚same' job offers from different ones. Quite often recruitment agencies post job offers that are virtually identical, with just one word of difference: the name of the position. After manually reviewing a number of offers and the results of the comparisons, it was found out that there is usually a range of values where the offers can be deemed ‚same' or ‚very similar'. It certainly is a fuzzy area, but in needs to be drawn in order to be able to assess the algorithms' efficiency.

For each of the algorithms the assessed threshold level means that when comparison result between two offers is above this level, there is an unacceptable (over 10 %) chance the offers will be different. Actually, the steepness of the curve

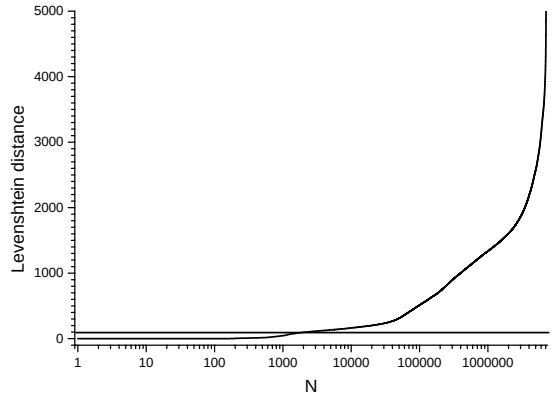

Fig. 1. Sorted results of Levenshtein distance for one to one comparisons of 3800 unique offers. Horizontal axis (N) represent comparison number and has a logarithmic scale

Rys. 1. Rezultaty (posortowane) porównania zbioru 3800 ofert przy pomocy metryki Levenshtein-a. Na osi poziomej (skala logarytmiczna) oznaczono numer porównania

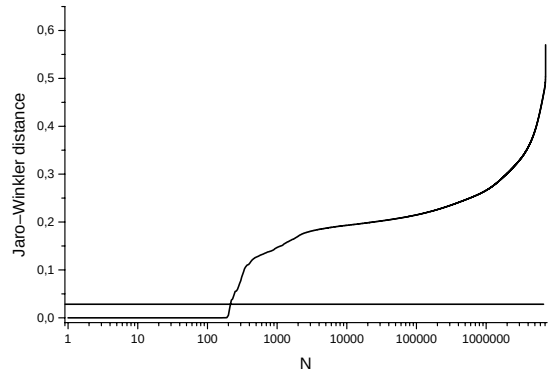

Fig. 2. Sorted results of Jaro-Winkler index for one to one comparisons of 3800 unique offers. Horizontal axis (N) represent comparison number and has a logarithmic scale

Rys. 2. Rezultaty (posortowane) porównania zbioru 3800 ofert przy pomocy indeksu Jaro-Winkler. Na osi poziomej (skala logarytmiczna) oznaczono numer porównania

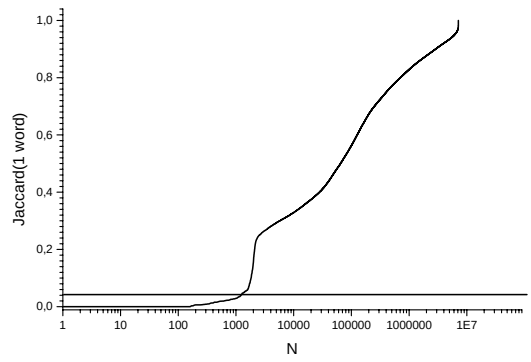

Fig. 3. Sorted results of Jaccard similarity measurement for one to one comparisons of 3800 unique offers. Horizontal axis (N) represent comparison number and has a logarithmic scale

Rys. 3. Rezultaty (posortowane) porównania zbioru 3800 ofert przy pomocy indeksu Jaccard-a. Na osi poziomej (skala logarytmiczna) oznaczono numer porównania

on fig. 1–3 while it crosses the threshold level denotes the reliability of the algorithm. The steeper (more vertical) the curve is, the better chance of correctly separating the similar and non-similar offers. As can be seen in fig. 1–3, Jaro-Winkler and Jaccard are much better at this than Levenshtein, providing higher certainty that the offers classified as 'same' really are such.

On the other hand, Levenshtein and Jaro-Winkler detect much fewer similar offers that Jaccard algorithm does. To confirm that, a smaller subset of offers were manually reviewed and were also subject to assessment by these algorithms. The results are presented in tab. 2.

Tab. 2. Algorithms' results against a manual review
Tab. 2. Rezultaty działania algorytmów

	Similar offers	False positives
Manual review	114	0
Levenshtein	73	0
Jaro-Winkler	63	0
Jaccard	112	2

To compare how the algorithms perform under the same conditions, the results of comparison of each pair by Levenshtein and Jaro-Winkler algorithms against the Jaccard algorithm. The results are presented in fig. 4–5 and clearly show that both algorithms, but especially Jaro-Winkler are more discriminative (steeper curve) at the beginning, where the differences between the texts compared are low.

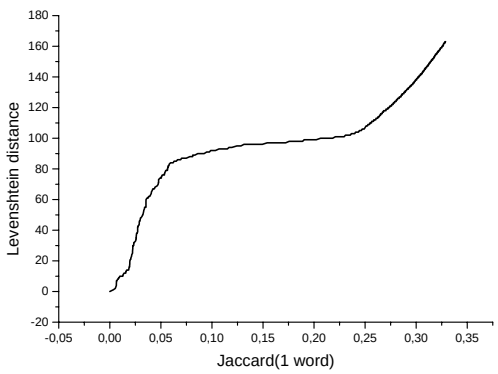

Fig. 4. Levensthein values against Jaccard similarity index for the same pairs of offers compared

Rys. 4. Wartość metryki Levenshtein-a w porównaniu do indeksu Jaccard-a dla tych samych par porównywanych ofert

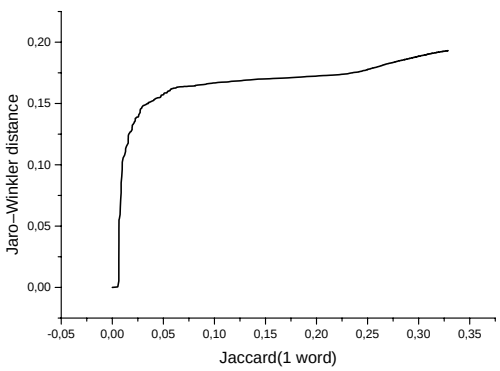

Fig. 5. Jaro-Winkler values against Jaccard similarity index for the same pairs of offers compared

Rys. 5. Wartość metryki Jaro-Winkler w porównaniu do indeksu Jaccard-a dla tych samych par porównywanych ofert

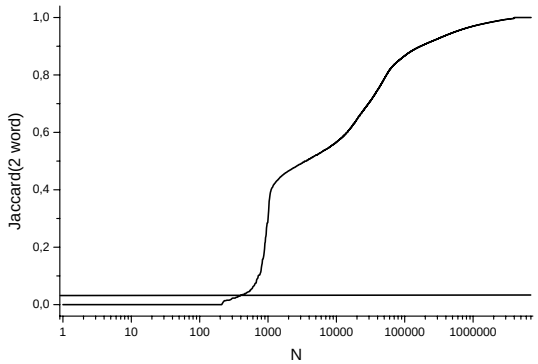

Fig. 6. Sorted results of Jaccard similarity measurement of 2-word clusters. Horizontal axis (N) represent comparison number and has a logarithmic scale

Rys. 6. Rezultaty (posortowane) porównania zbioru przy pomocy indeksu Jaccard-a pracującego na parach wyrazów. Na osi poziomej (skala logarytmiczna) oznaczono numer porównania

As mentioned before, Jaccard algorithm can compare documents split into arbitrary tokens (not only letters or words, but also n-grams; e.g. clusters of words). To verify the potential of this option, a test was run utilizing bi-words (documents split into two words clusters). The results are presented in Fig. 6. As can be seen, this gives much more discrimination, eliminating many similar offers. This is due to the fact that a change of one word in the whole text will offset the whole remaining text, therefore none of the following bi-words would match.

6. Conclusions

In this article, usability of certain text-comparison metrics for the purpose of identifying identical/similar job offers has been evaluated. Of the considered algorithm, Jaccard similarity index was found to be best suited for the task. This is likely due to the fact, that compared to e.g. Levenshtein distance that utilizes character distance, Jaccard similarity index was calculated on the basis of whole words, which better match this real-life scenario. At the same time, none of the algorithms were successful at differentiating job offers that came from the same recruitment agency and contained the same text, with one important difference; e.g. the job position. This seem an important issue that cannot effectively be tackled without actual understanding of the content and the importance of its certain parts or words.

More sophisticated algorithms that could possibly address the issue couldn't be used for this specific task as the number of offers registered in job sites amount to over a thousand a day. As a result, the number of comparisons (2-combination of the set) grows enormously (the offers must be compared across a range of days). Even limited, sub 4000 items sets required up to a day of computing time on a modern PC.

Bibliography

1. Jaccard P.: *Etude comparative de la distribution florale dans une portion des Alpes et des Jura*. "Societe Vaudoise des Sciences Naturelles", vol. 37, 547–579, 1901.
2. Levenshtein V.I.: *Binary codes capable of correcting deletions, insertions and reversals*, „Soviet Physics Doklady", vol. 10, 707–710, 1966.
3. Liu B., Ma Y., Yu P.: *Discovering Unexpected Information from Your Competitors' Web Sites*, Proceedings of ACM SIG KDD, 144–153, 2001.
4. Lukashenko R., Graudina V., Graudspenkis J.: *Computer-based plagiarism detection methods and tools: an overview*, Proceedings of Computer systems and technologies, 40:1–40:6, 2007.
5. Manning Ch. D., Schutze H.: *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
6. Myers G.: *A fast bit-vector algorithm for approximate string matching based on dynamic programming*, "Journal of the ACM", vol. 46(3), 395–415, 1999.
7. Navarro G.: *A guided tour to approximate string matching,* "ACM Computing Surveys" vol. 33(1), 31–88, 2001.
8. Winkler W.E.: *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunster Model of Record Linkage*, Proceedings of the Section on Survey Research Methods, 354–359, 1990. ■

Ocena skuteczności metryk porównywania tekstów dla potrzeb oceny liczby unikalnych ofert pracy

Streszczenie: W artykule przedstawione zostały rezultaty oceny możliwości zastosowań algorytmów porównywania tekstu dla potrzeb identyfikacji identycznych lub podobnych ogłoszeń o pracę. Do porównań wykorzystano klasyczne metryki (Levenshteina, Jaro-Winklera i Jaccarda). Oceniona została skuteczność i możliwość zastosowania tych algorytmów do przedstawionego zadania. Omówione zostały też kwestie analizy danych pobieranych ze stron www oraz niezbędnych nakładów obliczeniowych.

Słowa kluczowe: metryki tekstu, porównywanie tekstu, ocena ilości, automatyczna analiza stron www

dr inż. Marek Zachara

Adiunkt w Katedrze Automatyki Wydziału EAiIE AGH, konsultant w zakresie oceny bezpieczeństwa aplikacji internetowych. Zawodowo zajmuje się aspektami modelowania bezpieczeństwa, automatycznej analizy danych, testowania i kontroli wdrożenia.

http://marek.zachara.name
e-mail: mzachara@agh.edu.pl

mgr Cezary Piskor-Ignatowicz

Asystent w Katedrze Automatyki Wydziału EAiIE AGH. Ukończył Fizykę na Uniwersytecie Jagiellońskim. Obecnie zajmuje się analizą dynamiki kształtowania cen oraz bierze udział w projekcie mającym na celu stworzenie języka opisu funkcjonalności.

e-mail: ignatow@agh.edu.pl