2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Similarity Measures for Title Matching

Najlah Gali, Radu Mariescu-Istodor, Pasi Fränti
School of Computing
University of Eastern Finland
Joensuu, Finland
{najlaa, radum, franti}@cs.uef.fi

Abstract— **In many web applications, users query a place name, a photo name, and other entity names using search words that include alternate spellings, abbreviations, and variants that are similar, but not identical to the title associated with the desired entity. Given two titles, an effective similarity measure should be able to determine whether the titles represent the same entity or not. In this paper, we evaluate 21 measures with the aim of detecting the most appropriate measure for matching the titles. Results show that Soft-TFIDF performs the best.**

Keywords—similarity measures; title matching; web mining; information retrieval.

I. Introduction

The title is a descriptive name given to a book, an article, a document, an image or a web page. It summarizes the content and distinguishes it from other entities. The ability to accurately determine the similarity between the titles has an important impact on several text applications: topic detection, text mining, text summarization, query-answer applications, information retrieval, duplicate records identification, document clustering, and image retrieval.

A large number of methods have been developed to extract the titles of the documents; however, little attention has been given to which measure should be used to match the titles. Given two titles, an effective similarity measure should able to determine whether they are the same or not.

Computing the similarity between the titles is not a trivial task as it seems to. One reason is that the complexity of the structure of the title varies; a title can be a word, a phrase, or sentence with varying length. Another reason is that titles that describe the same entity might be syntactically different. For example, the same restaurant can be referred to as *Rosso* and *Rosso restaurant*. The title can also have different typographical/textual representation, for example:

- Fit for less – Fit4less
- Pizza at home – Pizza @ home
- Ching a lings– Ching-a-lings
- Ruby Lotel – Ruby L'otel.

Traditional similarity measures such as *cosine similarity* [1] may fail to correctly determine whether two titles are alike because of the variations in the representation. These variations are often found in the titles that are automatically extracted from unstructured, semi-structured documents, or web pages due to the difference in the formats. For example, re-ordering of the words or misspelling:

- Oliver Twist – Twist, Oliver
- QE Spa – Spa at QE
- Speech recognition – Speach recognition
- Business strategy – Business stragedy

One word can be more informative than another when comparing the titles for equivalence. For example, missing a descriptive word such as *Inc.* from the title *Lenovo Inc.* is not as important as missing the word *Lenovo*. Therefore, accurate similarity computation requires that the measure can compensate this kind of issues. Table I shows examples of titles that represent the same thing, and are therefore supposed to have high similarity, but for which the measures tend to give rather low scores, and titles that have problems from a human perspective, but for which the measures tend to give high scores.

Existing research have focused on several matching tasks:

- Short segments of text [2], *Apple computer –Apple pie*
- Sentences [3], *I haven't watched television for ages –It's been a long time since I watched television.*
- Named entities [4], [6], *U.S State Department – US Department of State.*
- Personal name [7], [8], *Gail Vest – Gayle Vesty.*
- Place name [9], *Ting Tsi River – Tingtze River.*
- Ontology alignments [10], *Associate professor –Senior lecturer.*

In [2] various similarity measures have been evaluated for the query-query similarity task, in which two short segments of text such as *MAC OS X* and *IMAC* is compared. Fourteen measures between two sentences were evaluated in [3]. Their main finding is that linguistic measures work better in identifying paraphrases than the word overlap and term frequency-inverse document frequency (TF-IDF).

TABLE I. EXAMPLES FOR TITLE MATCHING BY DIFFERENT MEASURES

Titles	**Measures**			
	Leven.	***Trigrams***	***Cosine***	***Jaccard***
Pizza Buffa Buffa Pizza	0.30	0.50	1.00	1.00
Microsoft Corporation Micrsoft Corporation	0.95	0.90	0.50	0.30
Ruby L'otel Ruby Lotel	0.90	0.80	0.50	0.30
Lenovo Inc. Lenovo	0.50	0.60	0.70	0.50
Infuzions W Infuzions	0.80	0.80	0.70	0.50
Out of the Blue Of the Blue	0.70	0.80	0.90	0.80

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

Several measures for matching the named entities have been compared in [4]. A new measure called soft-TFIDF was also introduced by combining the cosine distance with TF-IDF weighted vectors, and the Jaro-Winkler [5]. The authors conclude that soft-TFIDF works best for name matching. The soft-TFIDF was then extended in [6] by defining a family of similarity measures that combines edit-distance similarities and soft-TFIDF for matching named entities.

In [7] and [8] the performance of several similarity measures for personal name matching (first name, middle name, and sir name) are evaluated. Both studies have reported that it is hard to choose the best measure for this task. The same set of the measures reviewed in [7] has also been studied in [9] for matching the name of the places such as *Aldwincle Saint Peter* and *Saint Peter Aldwinkle*, but excluded measures that are based on phonetic encoding because of the language dependency. Their main finding is that Jaro-Winkler works best for places in China and Japan, q-grams for France, Germany, Italy, Mexico, Spain, and the United Kingdom, and edit-distance for Taiwan. A wide range of similarity measures for ontology alignment was studied in [10]. The authors reported that Soft-TFIDF, Jaccard, and Soft-Jaccard perform best for Biomedical, Soft-TFIDF for multilingual, and exact match, Jaccard, Levenshtein, q-grams, Soft-Jaccard, TF-IDF, and Soft-TFIDF for Standard English ontology, but not Monge-Elkan and Longest common substring (LCS).

All the studies point out that the performance of the similarity measures is affected by characteristics such as *text length*, *spelling accuracy*, *presence of abbreviations* and the *language*. Another common observation is that measures that demonstrate good performance and robustness for one data set can perform poorly on another.

Although string similarity measure is not a new area of research, it remains unclear which measure is useful for title matching. In this paper, we study 21 similarity measures with the aims at finding the best measure for matching the titles.

II. String Similarity Measures

Strings can be similar in two ways: *syntactically* when they share the same character sequence and *semantically* when they carry the same meaning (e.g., synonyms). In this work, we focus on the surface form to be independent from any language resources.

Similarity measures can be divided into four classes: *character-based*, *q-grams*, *token-based* and *mixed measures*. Character-based and q-grams measures calculate the similarity based on the sequence of the characters in the two strings. Token-based measures split the strings into words, and symbols (called tokens) using whitespace, line break or punctuation characters and then compute the similarity between the two token sets. Mixed measures combine the character- and token-based measures. Table II summarizes some of the measures we consider in this study. All measures are normalized to the scale [0, 1]; the closer the value to 1, the more similar the titles are.

A. Character-based measures

1) Edit distance: The minimum number of edit operations needed to transform a string *s* to a string *t*. The operations include insertion, deletion, and substitution of characters. Variants of the edit distance have been proposed depending on the number, type and the cost of the operations. *Hamming distance* allows only substitution, and the length of the two strings must be equal. *Levenshtein* [11] allows insertion, deletion, and substitution at a unit cost. *Damerau-Levenshtein* [12] has an additional operation of swapping two adjacent characters (*ab* ↔ *ba*) at cost 1. *Needleman-Wunsch* [13] uses non-uniform cost parameters for the basic edit operations. For example, cost of insertion and deletion is two, and cost of

TABLE II. SOME OF THE SIMILARITY MEASURES STUDIED

Name	Formula
Jaro [16]	$J(s,t) = \frac{1}{3} \times \left(\frac{c}{\lvert s\rvert} + \frac{c}{\lvert t\rvert} + \frac{x-c}{c}\right)$
Jaro-Winkler [5]	$JW(s,t) = J(s,t) + (l \times p(1 - J(s,t))$
Longest common substring [20]	$LCS(s,t) = \frac{L\acute{C}S(s,t)}{max(\lvert s\rvert, \lvert t\rvert)}$
Bi-Jaccard [18]	$Jac_{bi}(s,t) = \frac{\lvert bigr(s) \cap bigr(t)\rvert}{\lvert bigr(s) \cup bigr(t)\rvert}$
Bi-Dice [19]	$Dice_{bi}(s,t) = \frac{2 \times \lvert bigr(s) \cap bigr(t)\rvert}{\lvert bigr(s)\rvert + \lvert bigr(t)\rvert}$
Trigrams	$Trigram(s,t) \frac{trigr(s) \cap trigr(t)}{average(\lvert trigr(s)\rvert, \lvert trigr(t)\rvert)}$
Matching Coefficient	$MC(s,t) = \frac{\lvert s \cap t\rvert}{\max(\lvert s\rvert, \lvert t\rvert)}$
Overlap Coefficient	$OC(s,t) = \frac{\lvert s \cap t\rvert}{\min(\lvert s\rvert, \lvert t\rvert)}$
Jaccard	$Jac(s,t)_{token} = \frac{\lvert s \cap t\rvert}{\lvert s \cup t\rvert}$
Dice	$Dice(s,t)_{token} = \frac{2 \times \lvert s \cap t\rvert}{\lvert s\rvert + \lvert t\rvert}$
Rouge-N [21]	$F\,(s,t) = \left(\alpha \times \left(\frac{1}{p}\right) + (1-\alpha) \times \left(\frac{1}{r}\right)\right)^{-1}$ $\alpha = 0.5, \quad p = \frac{s \cap t}{\lvert s\rvert}, \quad r = \frac{s \cap t}{\lvert t\rvert}$
Cosine [1]	$\cos(s\,,t) = \frac{\sum_{i=1}^{\lvert\Sigma\rvert} s_i t_i}{\sqrt{\sum_{i=1}^{\lvert\Sigma\rvert}(s_i)^2}\ \sqrt{\sum_{i=1}^{\lvert\Sigma\rvert}(t_i)^2}}$
TF-IDF [22]	$sim_{tfidf}(s,t) = \sum_{w \in s \cap t} v(w,s) \times v(w,t)$ $\boldsymbol{v}(w,s) = \frac{\acute{v}(w,s)}{\sqrt{\sum_{w \in s} \acute{v}(w,s)^2}}$ $\acute{v}(w,s) = \log\left(TF_{w,s} + 1\right) \times \log(IDF_w)$ $\mathrm{TF}_{\mathrm{w,s}} = \mathrm{N}_{\mathrm{w,s}}\,, IDF_w = log \frac{\lvert str\rvert}{\lvert\{s \in T \mid w \in s\}\rvert}$
Euclidean [23]	$1 - \frac{\sqrt{\sum_{i=1}^{n}(s_i - t_i)^2}}{\sqrt{\lvert s\rvert^2 + (\lvert t\rvert)^2}}\,, \qquad n = s \cup t$
Manhattan [23]	$1 - \frac{\sum_{i=1}^{n} \lvert s_i - t_i\rvert}{\lvert s\rvert + \lvert t\rvert}\,, \qquad n = s \cup t$
Soft-TFIDF [4]	$Soft\,TFIDF(s,t) = \sum_{w \in CLOSE(\theta,s,t)} v(w,s) \times v(w,t) \times d(w,t)$ $CLOSE(\theta,s,t) = \{w \in s \mid \exists v \in t: s\acute{\imath}m(w,v) > \theta\}$ $\theta \geq 0.9$, $\mathrm{d(w,\,t)} = \max_{v \in t} s\acute{\imath}m(w,v)$
Monge-Elkan [24]	$ME\,(s,t) = \frac{1}{K}\sum_{i=1}^{k} \max_{j=1\ to\ L} s\acute{\imath}m\,(s_\imath\,, t_j)$ $\mathrm{K} = \lvert\mathrm{s}\rvert, \mathrm{L} = \lvert\mathrm{t}\rvert$

substitution is one. A variant of the Needleman-Wunsch called *Smith-Waterman* [14] focuses on local alignment by determining similar regions in the two strings. It assigns a lower cost when the mismatch happens at the beginning or at the end of the strings than when it happens in the middle. For example, given two strings "*Prof. Mohammed A. Gali, University of Baghdad*" and "*Mohammed A. Gali, Prof.*", the similarity between them using Smith-Waterman is 0.8, while it is 0.5 when using Needleman-Wunsch.

Smith-Waterman-Gotoh [15] improves the scaling of Smith-Waterman by adding a so-called *affine gap cost*. It introduces two costs for insertion: *gap open*, a penalty that corresponds to the beginning of a string of unmatched characters and *gap extension*, a penalty for its continuation. In addition, transformation between similar-sounding characters (e.g., {d, t}, {g, j}) is given a different weight from the match/mismatch weights. For example, five units are assigned for matching characters, three units for similar-sounding, and -3 for a mismatch.

2) Jaro distance [16] uses the number and the order of the common characters as follows: First, it computes the length of the two strings $|s|$ and $|t|$. Second, it finds the common characters (c) between the two strings; two characters match if they are the same and located no farther than $[max(|s|, |t|)/2]-1$ in the string. Third, it finds the number of transpositions ($x = m/2$), which is the number of matching characters (m), but in reverse order (a/u, u/a). *Winkler* [5] modified Jaro distance metric by adding a prefix scalier ($p = 0.1$) which gives higher weight to the strings that have a common prefix of length l up to four characters.

3) Longest common substring (LCS) [20] has been used in applications such as matching patient records in a clinical setting and text summarization, but not for comparing titles. We, therefore, study LCS for the title matching as well. It calculates the longest substring that co-occurs in both strings. The result is normalized by dividing it by the character-length of the longest string.

B. Q-grams

Count the number of substrings of length q that are common between the two strings [17]. The intuition is that the sequence of the characters is more important than the characters alone. Several measures have been introduced with a different value of q and different normalization. *Jaccard Index* [18] and *Dice Coefficient* [19] counts the number of common bigrams (2-grams); Jaccard divides the sum by the total number of unique bigrams in both strings while Dice divides it by the total number of bigrams in both strings. We also consider *Trigrams* (3-grams) divided by the average number of the trigrams in both strings.

Character-based and q-grams similarity measures work well for typographical errors. However, they fail to capture the similarity of the strings when the order changes (e.g., *Manta Café* versus *Café Manta*). Token-based measures try to compensate for this problem.

C. Token-based measures

Token-based measures convert the strings into tokens and discard the order in which the tokens occur in the two strings. For example, the string *computer science* is transformed to ("*computer*", "*science*") before the similarity is computed. The simplest form counts the number of tokens that both strings have in common, divided by the number of tokens in the longest string (*Matching coefficient*), the shortest string (*Overlap coefficient*), the total number of unique tokens in both strings (*Jaccard index*) or by the total number of all tokens in both strings (*Dice coefficient*).

Rouge-N stands for Recall-Oriented Understudy for Gisting Evaluation [21]. It uses *F-score* which combines *precision* (the number of common tokens divided by the number of tokens in the candidate string) and *recall* (the number of common tokens divided by the number of tokens in the ground truth string). We use Rouge-1 as was reported to work best for a very short text in [21].

More refined measures use feature vectors in which the tokens are represented by features such as occurrence and frequency in the string. *Cosine similarity* uses binary weighting (1 = occurrence; 0 = otherwise). For example, given strings s, t, x, and y, their binary feature vector is as follows:

- Theoretical computer science, $\vec{s} = (1, 1, 1)$;
- Computer science, $\vec{t} = (0, 1, 1)$;
- Computer, $\vec{x} = (0, 1, 0)$;
- Science, $\vec{y} = (0, 0, 1)$.

Tokens can also be represented by their TF-IDF. The term frequency ($TF_{w,s}$) is the number of times a token w occurs in s. The inverse document frequency (IDF_w) is the inverse of the number of strings (str) that contain w. The cosine measure with TF-IDF weighting is referred to as the *TF-IDF measure* [22]. *Euclidean distance and Manhattan distance* [23] use the frequency of the tokens in the string to generate the feature vector. For example, *The club at the Ivy* is represented by $\vec{s} = (2, 1, 1, 1)$. The distance between the vectors is then computed using the formulas in Table II.

Token-based measures consider two tokens match only if they are identical, but *computer* versus *computers* = mismatch. This kind of measures work well when the order of the tokens is not important, but they fail with the strings that are slightly different (e.g., *color* versus *colour*). Therefore, more flexible measures are needed.

D. Mixed measures

The principle of the mixed measures is to apply a character-based measure (secondary measure) to all pairs of tokens between the two strings and consider only tokens that satisfy a certain criterion (e.g., threshold) as input to a token-based measure. *Monge-Elkan* [24] takes the average score of the best matching tokens from the secondary measure such as Levenshtein, Jaro, and Smith-Waterman. We use here, Smith-Waterman-Gotoh as the secondary measure.

Soft-TFIDF combines the TF-IDF and Jaro-Winkler measures. It first applies Jaro-Winkler ($s\acute{\imath}m$) to all pairs of tokens between the two strings, and then applies the TF-IDF measure to tokens that have a similarity score above the threshold ($\theta \geq 0.9$) according to Jaro-Winkler.

III. Experiments

We compare 21 similarity measures. Fifteen of them are implemented in SimMetrics[1], which is an open source Java library provided by UK Sheffield University. TF-IDF and Soft TF-IDF are from SecondString[2] Java toolkit [4], Rouge-1 is from [21], and Hamming, Damerau-Levenshtein, LCS and the bigrams Dice and Jaccard measures are implemented by us. We conducted six experiments with different set up:

- Text manipulation (char. change, token change, token swap)
- Correlation to human judgments
- Correlation to distance
- Clustering

A. Data Sets

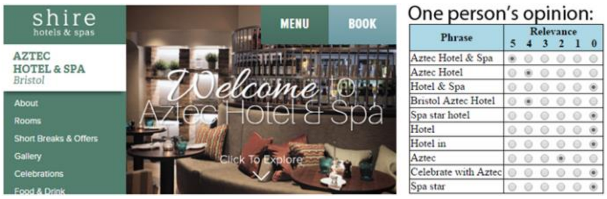
We use two types of data sets: *Titler*[3] [25] and *Mopsi photo collection*[4]. Titler data set contains 4,968 candidate title phrase extracted from 1,002 English websites, and the ground truth titles that were manually extracted by two persons independently of each other. In the case of disagreement, a third person made a judgment between the two. The candidate title phrases were extracted automatically from the pages using the method developed in [25] and evaluated for their relevance by the humans using *TitleRater tool* shown in Fig.1. The user rates the candidates from 0 (irrelevant) to 5 (excellent match). Scales 1 to 4 represent the difference of the human ratings.

Mopsi photo collection contains 42,739 geotagged photos by April 2016. Each photo may have a short description (English or Finnish). We conducted the experiment on 1000 most recent photos that have a description. Table III summarizes the specifications of the data sets.

B. Text manipulation

We first examined how each measure performs under manipulation: character change, token change, and reordering of tokens. We selected *Speech and image processing unit* as a baseline and applied several systematic changes. We first executed k character changes (k = 1 to 32) and then k token changes (k = 1 to 6). For each value of k, we performed 90 and 35 iterations of characters and tokens change respectively, and reported the average result. We also examined the effect of changing the order of the tokens by swapping them 100 times and average the result. In Fig. 2, the majority of the character-based measures show a constant decrease of similarity, but the amount of decrease varies. The modified versions of the Levenshtein such as Damerau, Needleman-Wunsch, and Smith-Waterman correlate best with the amount of error added to the text (explained to be expected result).

Q-grams measures show a uniform decrease with the number of characters being changed. They are more sensitive to the character changes because these will destroy the bi- and trigrams. The most noticeable exception is the Bi-Jaccard, which performs as well as Jaro-Winkler. The mixed measures generally drop faster than the character-based measures because they discard an entire token if its matching score is below the threshold, but Monge-Elkan and Soft-TFIDF still do better than Levenshtein, Smith-Waterman-Gotoh, LCS, and q-grams. In conclusion, measures that depend on a single character edit are closer to the expectation than q-gram based measures such as Bi-Dice and Trigrams. The exceptions are Hamming, Levenshtein, Smith-Waterman-Gotoh, Jaro and LCS measures.



One person's opinion:

Phrase	Relevance 5	4	3	2	1	0
Aztec Hotel & Spa	●	○	○	○	○	○
Aztec Hotel	○	●	○	○	○	○
Hotel & Spa	○	○	○	○	○	●
Bristol Aztec Hotel	○	●	○	○	○	○
Spa star hotel	○	○	○	○	○	●
Hotel	○	○	○	○	○	●
Hotel in	○	○	○	○	○	●
Aztec	○	○	○	●	○	○
Celebrate with Aztec	○	○	○	○	○	●
Spa star	○	○	○	○	○	●

Fig. 1. Evaluation tool for human ratings.

TABLE III. SUMMARY OF DATA SETS

Data set	Size	Type	Language	Length of title
Titler	4968	Title phrases	English	Min= 1, Max= 6, Av.= 2
Mopsi photo	1000	Photo descriptions	English Finnish	Min= 1, Max= 11, Av.= 2

Fig. 3 illustrates that token-based measures have uniform decrease with respect to the token change, except Euclidean distance which converges to 0.5. Matching coefficient, Overlap, Dice, Cosine, and Manhattan decrease the same amount as the number of tokens being changed. As assumed, Monge-Elkan provides slightly higher similarity score than the expected due to its ability to capture the similarity between similar and identical tokens. TF-IDF, Soft-TFIDF, Jaccard and Rouge-1 provide almost similar scores, although Soft-TFIDF was expected to perform as Monge-Elkan, on average.

Table IV shows that none of the token-based measures is affected by the change of the order of the tokens. Among the q-grams, Bi-Jaccard and Bi-Dice perform as well as the token-based measures because they match the bigrams regardless of their positions in the strings. None of the character-based measures performs well when swapping the order of the tokens, but Needleman-Wunsch gives better results, and Hamming is mostly affected by the order change.

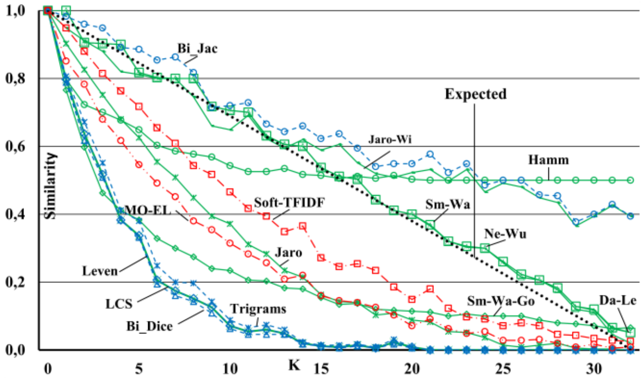


Fig. 2. Effect of character changes on character-based, q-grams and mixed measures.

[1] https://sourceforge.net/projects/simmetrics/

[2] https://sourceforge.net/projects/secondstring/

[3] http://cs.uef.fi/mopsi/titler/

[4] http://cs.uef.fi/mopsi/tools/photoclusters.php

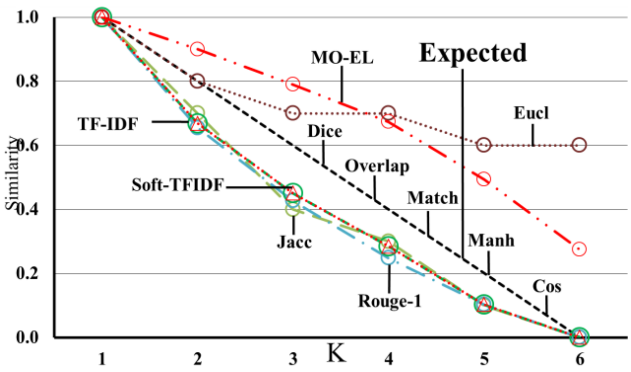


Fig. 3. Effect of token changes on token-based and mixed measures.

C. *Correlation to human judgments*

Next, we use Titler data set to detect how well the similarity scores provided by different measures correlate with the human ratings. For non-symmetric measures such as Monge-Elkan, we use the F-score in the comparison. From Table IV we observe that all measures have a positive correlation to the human judgments. The strength of the correlation is moderate (from 0.44 to 0.59) for most measures. The correlation of Levenshtein, Damerau-Levenshtein, Needleman-Wunsch, LCS, and Trigrams is slightly higher (0.56 – 0.59) than that of the others (0.44 – 0.53). Smith-Waterman, Smith-Waterman-Gotoh, and the Overlap have a weak correlation (0.21 – 0.25). Monge-Elkan improves the performance of the Smith-Waterman-Gotoh and Soft-TFIDF performs better than Jaro-Winkler and TF-IDF alone. This indicates that the combination of a character-based measure and a token-based measure correlates better to the human judgments than the measure alone such as Soft-TFIDF (0.51) versus Jaro-Winkler (0.39) and TF-IDF (0.47).

To investigate why none of the measures strongly correlate with the human judgments, we analyzed the ratings further, and we observed that users pay less attention to the typographical difference between the ground truth titles and the candidate phrases as they consider these phrases an excellent match, for example:

- Freda's – Fredas
- Drom UK – Dröm UK
- Hot Spring – HotSpring
- Park Hotel and Spa – Park Hotel & Spa

We further observed that phrases that miss some less relevant descriptive information or having additional information were still rated high by humans. For example, all the following pairs were given score 4:

- Paradox – The Paradox
- De La Esquina – Café De La Esquina
- Lunckmam Park Hotel and Spa – Lunckman Park

Furthermore, the measures provide high similarity when matching the following phrases, but the humans did not rate them as highly relevant:

- Out of the Blue – Out the Blue
- Arcata Pizzeria – At Arcata Pizzeria
- 3 Degrees – Degrees

All these have a significant impact on the degree of correlation between the measures and the human ratings.

D. *Correlation to distance*

Two photos taken in the same location are expected to have the same (or at least more similar) description more often than two photos in different (random) locations. Accordingly, for a given input photo, a similarity measure should rank the nearby photo similar more often than the far-away photo. We test this hypothesis by counting the number of times it happens. Expected result for two random is 50 %. The result is not expected to reach 100 % since not all nearby photos describe the same object. However, good similarity measure should provide higher values since the process is otherwise completely random.

We use 1000 randomly selected Mopsi photos to examine how much the similarity measures correlate with the assumption that nearby photos are more likely to have a similar description.

Table IV shows that Euclidean provides the highest similarity scores between the nearby photos. It also has good correlation with the human judgments. Likewise, Levenshtein, Damerau-Levenshtein, and Needleman-Wunsch correlate best with the human judgments, and they provide high similarity scores between the photos. This indicates that measures that correlate with the hypothesis that nearby photos share similar description also correlate with the human judgments. In general, character-based measures perform better than token-based measures for the nearby photos. This is due to the fact that token-based measures fail to capture the similarity between similar tokens with some artifacts; therefore they would give the same score to "*snow hotel*" versus "*snow hoteli*" and "*snow hotel*" versus "*snow football*".

E. *Clustering*

We further tested the goodness of the measures by clustering the Mopsi photos. We manually selected 180 photos having common strings describing the same object. We then checked whether these photos are partitioned into the same clusters. In testing, we clustered all 1000 photos into 100 groups using the different similarity measures and then counted the number of times the selected photos are found in a different cluster. The final result is divided by the number of all matching photos.

In Table IV we observe that token-based measures perform better in this clustering scenario. In contrast, character-based measures perform better for matching the nearby photos. Exceptions are Smith-Waterman and Trigrams; which give good clustering probably because of the word merging characteristic of the Finnish language. Table IV also shows that no measure outperforms others in all experiments. Character-based measures correlate best with the human judgments and perform well for matching the nearby photos, but they are outperformed by the token-based and the mixed measures in finding good clusters. Mixed measures also have good properties and correlate well with the human judgments. This indicates that a good combination of a character- and a token-based measure might work best for the title matching task.

IV. CONCLUSION

Considering the behavior of the measures within own class, the correlation to the human judgments for Titler data set, the similarity between the nearby photos, and clustering similar photos in Mopsi photo collection, we conclude that among the character-based measures Damerau-Levenshtein, Needleman-Wunsch, and Smith-Waterman perform well under character changes, but only Damerau-Levenshtein and Smith-Waterman for token changes. Damerau-Levenshtein works also well with the real data, and would probably be the best choice.

Most q-gram measures are poor with character changes but works well with token changes and real data. Only Bi-Jaccard works reasonably for both and also for the real data. Trigram works well for all except character changes. If this is not critical, it might be the one to recommend. As expected, all token-based measures are invariant to token swaps and most to token changes, but they all are vulnerable to character changes. Rouge-1 and Dice are less affected.

Mixed measures manage to combine the best properties of the character- and token-based measures. But the studied combinations are clearly not the best ones, so it is expected that better combination can be found from Damerau-Levenshtein and Dice, for example. Our future work will focus on this direction.

TABLE IV. SUMMARY OF THE SIX EXPERIMENTS: GREEN (EXCELLENT), BLUE (GOOD), AND RED (POOR).

	Text manipulation			Titler	Mopsi photo	
	Char. change	Token change	Token swap	Corr. to human	Corr. to dist.	Clus.
Character-based						
Hamming	0.20	🟦 0.05	🟥 0.14	-	61	🟥 0.19
Levenshtein	🟥 0.36	🟦 0.07	0.39	🟩 0.59	🟦 76	0.11
Damerau- Leven.	🟩 0.02	🟦 0.07	0.39	🟩 0.59	🟦 76	0.11
Needleman- Wu.	🟩 0.02	🟥 0.28	0.62	🟩 0.56	🟦 77	0.11
Smith-Waterman	🟩 0.02	🟦 0.05	0.51	🟥 0.25	60	🟩 0.04
Smith-Wat.- Goto.	0.28	🟦 0.04	0.57	🟥 0.25	63	🟦 0.07
Jaro	0.23	0.16	0.58	0.45	71	🟥 0.16
Jaro-Winkler	🟦 0.13	0.22	0.60	0.39	71	🟥 0.14
LCS	🟥 0.36	🟦 0.07	0.43	🟩 0.56	66	0.11
Q-grams						
Bi-Jaccard	🟦 0.15	0.10	🟩 1.00	🟦 0.52	68	🟦 0.09
Bi-Dice	🟥 0.37	🟦 0.01	🟩 1.00	0.47	68	0.11
Trigrams	🟥 0.35	🟦 0.02	🟦 0.75	🟩 0.58	69	🟩 0.05
Token-based						
Matching	🟥 0.40	🟩 0.00	🟩 1.00	🟦 0.52	65	0.12
Overlap	🟥 0.36	🟩 0.00	🟩 1.00	🟥 0.21	64	🟦 0.08
Jaccard	0.26	🟦 0.08	🟩 1.00	🟦 0.53	64	🟦 0.07
Dice	🟦 0.16	🟩 0.00	🟩 1.00	0.45	64	🟦 0.06
Rouge-1	🟦 0.11	🟦 0.09	🟩 1.00	0.47	66	🟦 0.09
Cosine	0.31	🟩 0.00	🟩 1.00	0.44	64	0.10
TF-IDF	🟥 0.35	🟦 0.08	🟩 1.00	0.47	64	🟦 0.09
Euclidean	🟥 0.37	0.23	🟩 1.00	🟦 0.51	🟩 84	🟦 0.08
Manhattan	🟥 0.36	🟩 0.00	🟩 1.00	0.45	64	🟦 0.06
Mixed						
Monge-Elkan	0.24	0.19	🟩 1.00	🟦 0.50	60	🟦 0.08
Soft-TFIDF	🟦 0.15	🟦 0.08	🟩 1.00	🟦 0.51	65	🟦 0.07

REFERENCES

[1] W. Cohen, P .Ravikumar, and S. Fienberg. A comparison of string metrics for matching names and records. In Kdd workshop on data cleaning and object consolidation , 3, 73-78, August 2003.

[2] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. Springer Berlin Heidelberg, 16-27, April 2007.

[3] P. Achananuparp, X. Hu, and X. Shen. The evaluation of sentence similarity measures. In Data warehousing and knowledge discovery, 305-316, September 2008.

[4] W. Cohen, PD. Ravikumar, and SE. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In II Web, 73-78, 2003.

[5] WE. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, 1990.

[6] E. Moreau, F. Yvon, and O. Cappé. Robust similarity measures for named entities matching. Int. Conf. on Computational Linguistics, 1, 593-600, 2008.

[7] P. Christen. A comparison of personal name matching: Techniques and practical issues. In Data Mining Workshops. IEEE Int. Conf., 290-294, December 2006.

[8] C. Snae. A comparison and analysis of name matching algorithms. Int. Journal of Applied Science. Engineering and Technology., 4 (1), 252-257, January 2007.

[9] G. Recchia, and MM. Louwerse. A Comparison of String Similarity Measures for Toponym Matching. In COMP@ SIGSPATIAL, 54-61, 2013.

[10] M. Cheatham, Hitzler P. String similarity metrics for ontology alignment. In the Semantic Web–ISWC 2013, 294-309, 2013.

[11] VI. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. InSoviet physics doklady,10 (8), 707-710, February 1966.

[12] FJ. Damerau. A technique for computer detection and correction of spelling errors. Communications of the ACM, 7 (3), 171-176, 1964.

[13] SB. Needleman, and CD. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48 (3), 443-53, March 1970.

[14] TF Smith, and MS.Waterman. Identification of common molecular subsequences. Journal of molecular biology, 147 (1), 195-197, 1981.

[15] O. Gotoh. An improved algorithm for matching biological sequences. Journal of molecular biology, 162 (3), 705-8, December 1982.

[16] MA. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association. 84 (406), 414-420, June 1989.

[17] K. Kukich. Techniques for automatically correcting words in text. ACM Computing Surveys, 24 (4), 377-439, 1992.

[18] P. Jaccard Etude comparative de la distribution florale dans une portion des Alpes et du Jura. Bulletin de la societe Vaudoise des Sciences Naturelles, 37, 547-579, 1901.

[19] C. Brew, and D. McKelvie. Word-pair extraction for lexicography. Int. Conf. on New Methods in Language Processing, 45-55, 1996.

[20] C. Friedman, R. Sideli. Tolerating spelling errors during patient validation. Computers and Biomedical Research,31 (5), 486-509, 1992.

[21] C. Y. Lin. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: The ACL-04 workshop, 8, 2004.

[22] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel Similarity measures for tracking information flow. ACM int. conf. on Information and knowledge management, 517-524, 2005.

[23] P. Malakasiotis, and I. Androutsopoulos. Learning textual entailment using SVMs and string similarity measures. The ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, 42-47, 2007.

[24] A. E.Monge, and C. Elkan. The Field Matching Problem: Algorithms and Applications. Int. Conf. on Knowledge Discovery and Data Mining, 267-270,1996.

[25] N. Gali, R. Mariescu-Istodor and P. Fränti. Learning Based Method for Web Title Extraction Using Linguistic Knowledge. Manuscript, 2016.