Robust Similarity Measures for Named Entities Matching

Erwan Moreau[1]
Institut Télécom ParisTech
& LTCI CNRS
erwan.moreau@enst.fr

François Yvon
Univ. Paris Sud
& LIMSI CNRS
yvon@limsi.fr

Olivier Cappé
Institut Télécom ParisTech
& LTCI CNRS
cappe@enst.fr

Abstract

Matching coreferent named entities without prior knowledge requires good similarity measures. Soft-TFIDF is a fine-grained measure which performs well in this task. We propose to enhance this kind of metrics, through a generic model in which measures may be mixed, and show experimentally the relevance of this approach.

1 Introduction

In this paper, we study the problem of matching coreferent named entities (NE in short) in text collections, focusing primarily on orthographic variations in nominal groups (we do not handle the case of pronominal references). Identifying textual variations in entities is useful in many text mining and/or information retrieval tasks (see for example (Pouliquen et al., 2006)). As described in the literature (e.g. (Christen, 2006)), textual differences between entities are due to various reasons: typographical errors, names written in different ways (with/without first name/title, etc.), abbreviations, lack of precision in organization names, transliterations, etc. For example, one wants *"Mr. Rumyantsev"* to match with *"Alexander Rumyanstev"* but not with *"Mr. Ryabev"*. Here we do not address the related problem of disambiguation[2] (e.g. knowing whether a given occurrence of *"George Bush"* refers to the 41st or 43rd president of the USA), because it is technically very different from the matching problem.

There are different ways to tackle the problem of NE matching: the first and certainly most reliable one consists in studying the specific features of the data, and then use any available tool to design a specialized method for the matching task. This approach will generally take advantage of language-specific (e.g. in (Freeman et al., 2006)) and domain-specific knowledge, of any external resources (e.g. database, names dictionaries, etc.), and of any information about the entities to process, e.g. their type (person name, organization, etc.), or internal structure (e.g. in (Prager et al., 2007)). In such an in-depth approach, supervised learning is helpful: it has been used for example in a database context[3] in (Bilenko et al., 2003), but this approach requires labeled data which is usually costly. All those data specific appproaches would necessitate some sort of human expertise.

The second approach is the *robust* one: we propose here to try to match any kind of NE, extracted from "real world" (potentially noisy) sources, without any kind of prior knowledge[4]. One looks for coreferent NE, whatever their type, source, language[5] or quality[6]. Such robust similarity methods may be useful for a lot of generic tasks, in which maximum accuracy is not the main criterion, or simply where the required resources are not available.

The literature on string comparison metrics is abundant, containing both general techniques and

[1] Now at LIPN - Univ. Paris 13 & UMR CNRS 7030.

[2] Which is essential in the Web People Search task.

[3] The matching task is quite different in this framework, because one observes records (structured information).

[4] In this kind of knowledge are included the need for hand-tuning parameters or defining specific thresholds.

[5] Actually we have only studied English and French (our approach is neither "multilingual", in the sense that it is not specific to multilingual documents).

[6] In particular, this task clearly depends on the NE recognition step, which may introduce errors.

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 593–600
Manchester, August 2008

more linguistically motivated measures, see e.g. (Cohen et al., 2003) for a review. From a bird's eye view, these measures can be sorted in two classes: "Sequential character-based methods" and "Bag-of-words methods"[7]. Both classes show relevant results, but do not capture the same kind of similarity. In a robust approach for NE matching, one needs a more fine-grained method, which performs at least as well as bag-of-words methods, without ignoring coreferent pairs that such methods miss.

A first attempt in this direction was introduced in (Cohen et al., 2003), in the form of a measure called *Soft-TFIDF*. We will show that this measure has theoretical pitfalls and a few practical drawbacks. Nevertheless, Soft-TFIDF outperforms the better standard string similarity measures in the NE matching task. That is why we propose to generalize and improve its principle, and show experimentally that this approach is relevant.

In section 2 we introduce standard similarity measures and enhance the definition of Soft-TFIDF. Then we define a generic model in which similarity measures may be combined (section 3). Finally, section 4 shows that experiments with two different corpora validate our approach.

2 Approximate matching methods

We present below some of the main string similarity measures used to match named entities (Christen, 2006; Cohen et al., 2003; Bilenko et al., 2003).

2.1 Classical metrics

2.1.1 Sequential character based methods

Levenshtein edit distance. This well-known distance metric d represents the minimum number of insertions, deletions or substitutions needed to transform a string x into another string y. For example, $d(kitten, sitting) = 3$ ($k \mapsto s$, $e \mapsto i$, $\varepsilon \mapsto g$). The corresponding normalized similarity measure is defined as $s = 1 - d/max(|x|, |y|)$. A lot of variants and/or improvements exist (Navarro, 2001), among which:

- *Damerau.* One basic edit operation is added: a *transposition* consists in swapping two characters;
- *Needleman-Wunch.* Basic edit operation costs are parameterized: G is the cost of a gap (insertion or deletion), and there is a function $cost(c, c')$ which gives the cost of substituting c with c' for any pair of characters (c, c').

Jaro metric (Winkler, 1999). This measure is based on the number and the order of common characters. Given two strings $x = a_1 \dots a_n$ and $y = b_1 \dots b_m$, let $H = \min(n, m)/2$: a_i is *in common* with y if there exists b_j in y such that $a_i = b_j$ and $i - H \leq j \leq i + H$. Let $x' = a'_1 \dots a'_{n'}$ (resp. $y' = b'_1 \dots b'_{m'}$) be the sequence of characters from x (resp. y) in common with y (resp. x), in the order they appear in x (resp. y). Any position i such that $a'_i \neq b'_i$ is called a *transposition*. Let T be the number of transpositions between x' and y' divided by 2:

$$Jaro(x, y) = \frac{1}{3} \times \left(\frac{|x'|}{|x|} + \frac{|y'|}{|y|} + \frac{|y'| - T}{|y'|} \right)$$

2.1.2 Bag-of-words methods

With these methods, each NE is represented as a set of *features* (generally words or characters n-grams[8]). Let $X = \{x_i\}_{1 \leq i \leq n}$ and $Y = \{y_i\}_{1 \leq i \leq m}$ be the sets representing the entities x, y. Simplest measures only count the number of elements in common[9], e.g:

$$Overlap(x, y) = \frac{|X \cap Y|}{min(|X|, |Y|)}$$

Some more subtle techniques are based on a vector representation of entities x and y, which may take into account parameters that are are not included in the sets themselves. Let $A = (a_1, \dots, a_{|\Sigma|})$ and $B = (b_1, \dots, b_{|\Sigma|})$ be such vectors[10], the widely used cosine similarity is:

$$cos(A, B) = \frac{\sum_{i=1}^{|\Sigma|} a_i b_i}{\sqrt{\sum_{i=1}^{|\Sigma|} a_i^2} \sqrt{\sum_{i=1}^{|\Sigma|} b_i^2}}$$

Traditionally, TF-IDF weights are used in vectors (*Term Frequency-Inverse Document Frequency*). In the NE case, this value represents the importance each feature w (e.g. word) has for an entity x belonging to the set E of entities:

$$\mathrm{tf}(w, x) = \frac{n_{w,x}}{\sum_{w' \in \Sigma} n_{w',x}}, \mathrm{idf}(w) = \log \frac{|E|}{|\{x \in E | w \in x\}|},$$

$$\mathrm{tfidf}(w, x) = \mathrm{tf}(w, x) \times \mathrm{idf}(w).$$

with $n_{w,x}$ the number of times w appears in x. Thus the similarity score is $\mathrm{CosTFIDF}(x, y) = Cos(A, B)$, where each a_i (resp. b_i) in A (resp. in B) is $\mathrm{tfidf}(w_i, x)$ (resp. $\mathrm{tfidf}(w_i, y)$).

[7] We omit measures based on phonetic similarity such as Soundex, because they are language-specific and/or type-specific (person names).

[8] In the remaining the term *n-grams* is always used for *characters n-grams*.

[9] $|E|$ denotes the number of elements in E.

[10] Σ is the vocabulary, containing all possible features.

2.2 Special measures for NE matching

Experiments show that sequential character-based measures catch mainly coreferent pairs of long NE that differ only by a few characters. Bag-of-words methods suit better to the NE matching problem, since they are more flexible about word order and position. But a lot of coreferent pairs can not be identified by such measures, because of small differences between words: for example, *"Director ElBaradei"* and *"Director-General ElBareidi"* is out of reach for such methods. That is why "second level" measures are relevant: their principle is to apply a sub-measure sim' to all pairs of words between the two NE and to compute a final score based on these values. This approach is possible because NE generally contain only a few words.

Monge-Elkan measure belongs to this category: it simply computes the average of the better pairs of words according to the sub-measure:

$$sim(x,y) = \frac{1}{n}\sum_{i=1}^{n} \max_{j=1}^{m}(sim'(x_i, y_j)).$$

But experiments show that Monge-Elkan does not perform well. Actually, its very simple behavior favors too much short entities, because averaging penalizes a lot every non-matching word.

A more elaborated measure is proposed in (Cohen et al., 2003): *Soft-TFIDF* is intended precisely to take advantage of the good results obtained with Cosine/TFIDF, without automatically discarding words which are not strictly identical. The original definition is the following: let $\mathrm{CLOSE}(\theta, X, Y)$ be the set of words $w \in X$ such that there exists a word $v \in Y$ such that $sim'(w,v) > \theta$. Let $N(w,Y) = \max(\{sim'(w,v) | v \in Y\})$. For any $w \in \mathrm{CLOSE}(\theta, X, Y)$, let

$$S_{w,X,Y} = weight(w,X) \cdot weight(w,Y) \cdot N(w,Y),$$

where $weight(w,Z) = \dfrac{\mathrm{tfidf}(w,Z)}{\sqrt{\sum_{w \in Z} \mathrm{tfidf}(w,Z)^2}}$.

Finally,

$$\mathrm{SoftTFIDF}(X,Y) = \sum_{w \in \mathrm{CLOSE}(\theta,X,Y)} S_{w,X,Y}.$$

This definition is not entirely correct, because $weight(w,Y) = 0$ if $w \notin Y$ (in other words, w must appear in both X and Y, thus $\mathrm{SoftTFIDF}(X,Y)$ would always be equal to $\mathrm{CosTFIDF}(X,Y)$). We propose instead the following corrected definition, which corresponds to the implementation the authors provided in the package SecondString[11]:

Let $\mathrm{CLOSEST}(\theta, w, Z) = \{v \in Z \mid \forall v' \in Z : sim'(w,v) \geq sim'(w,v') \wedge sim'(w,v) > \theta\}$.

$$\mathrm{SoftTFIDF}(X,Y) = \sum_{w \in X} weight(w,X) \cdot \alpha_{w,Y},$$

where $\alpha_{w,Z} = 0$ if $\mathrm{CLOSEST}(\theta, w, Z) = \emptyset$, and $\alpha_{w,Z} = weight(w', Z) \cdot sim'(w, w')$ otherwise, with[12] $w' \in \mathrm{CLOSEST}(\theta, w, Z)$.

As one may see, SoftTFIDF relies on the same principle than Monge-Elkan: for each word x_i in the first entity, find a word y_j in the second one that maximizes $sim'(x_i, y_j)$. Therefore, these measures have both the drawback not to be symmetric. Furthermore, there is another theoretical pitfall with SoftTFIDF: in Monge-Elkan, the final score is simply normalized in $[0,1]$ using the average among words of the first entity. According to the principle of the Cosine angle of TF-IDF-weighted vectors, SoftTFIDF uses both vectors norms. However the way words are "approximately matched" does not forbid the matching of a given word in the second entity twice: in this case, normalization is wrong because this word is counted only once in the norm of the second vector. Consequently there is a potential overflow: actually it is not hard to find simple examples where the final score is greater than 1, even if this case is unlikely with real NE and a high threshold θ.

3 Generalizing Soft-TFIDF

3.1 A unifying framework for similarity measures

We propose to formalize similarity measures in the generic model below. This model is intended to define, compare and possibly mix different kinds of measures. The underlying idea is simply that most measures may be viewed as a process following different steps: representation as a sequence of features[13] (e.g. tokenization), alignment and a way to compute the final score. We propose to define a similarity measure sim through these three steps, each of them is modeled as a function[14]:

Representation. Given a set F of *features*, let $\mathrm{features}(e) = \langle a_1, \ldots, a_n \rangle$ be a function that as-

[11] `http://secondstring.sourceforge.net`

[12] If $|\mathrm{CLOSEST}(\theta, w, Z)| > 1$, pick any such w' in the set. In the case of matching words between NE, this should almost never happen.

[13] We use the word *feature* for the sake of generality.

[14] Of course, alternative definitions may be relevant. In particular one may wish to allow the alignment function to return a set of graphs instead of only one. In the same way, one may wish to add a special vertex ε to the graph, in order to represent the fact that a feature is not matched by adding an edge between this feature and ε.

signs an (ordered) sequence of features to any entity e ($a_i \in F$ for any i). Features may be of any kind (e.g. characters, words, n-grams, or even contextual elements of the entity) ;

Alignment. Given a function $sim^F : F^2 \mapsto \mathbb{R}$ which defines similarity between any pair of features, let $align(\langle a_1, \ldots, a_n \rangle, \langle a'_1, \ldots, a'_{n'} \rangle) = G$ be a function which assigns a graph G to any pair of features sequences. $G = (V, E)$ is a bipartite weighted graph where:

- The set of vertices is $V = A \cup A'$, where A and A' are the partitions defined as $A = \{v_1, \ldots, v_n\}$ and $A' = \{v'_1, \ldots, v'_{n'}\}$. Each v_i (resp. v'_i) represents (the position of) the corresponding feature a_i (resp. a'_i) ;
- The set of weighted edges is $E = \{(v_{i_j}, v'_{i'_j}, s_j)\}_{1 \leq j \leq |E|}$, where $v_{i_j} \in A$, $v'_{i'_j} \in A'$. Weights s_j generally depend on $sim^F(a_{i_j}, a'_{i'_j})$.

Scoring. Finally $sim = score(G)$, where $score$ assigns a real value (possibly normalized in $[0, 1]$) to the alignment G.

The representation step is not particularly original, since different kinds of representation have already been used both with sequential methods and "bag-of-features" methods. However our model also entails an alignment step, which does not exist with bag-of-features methods. Actually, the alignment is implicit with such methods, and we will show that making it visible is essential in the case of NE matching.

In the remaining of this paper we will only consider normalized metrics (scores belong to $[0, 1]$).

3.2 Revisiting classical similarity measures

Measures presented in section 2 may be defined within the model presented above. This modelization is only intended to provide a theoretical viewpoint on the measures: for all practical purposes, standard implementations are clearly more efficient. Below we do not detail the representation step, because there is no difficulty with it, and also because it is interesting to consider that any measure may be used with different kinds of features, as we will show in the next section. Let $S = \langle a_1, \ldots, a_n \rangle = \text{features}(e)$ and $S' = \langle a'_1, \ldots, a'_{n'} \rangle = \text{features}(e')$ for any pair of entities (e, e').

3.2.1 Levenshtein-like similarity

The function $align_{lev}(S, S')$ is defined in the following way: let $\mathcal{G}_{lev}$ be the set of all graphs $G = (V, E)$ such that any pair of edges $(v_{i_j}, v'_{i'_j}, s_j), (v_{i_k}, v'_{i'_k}, s_k) \in E$ satisfies $(i_j < i_k \wedge i'_j < i'_k) \vee (i_j > i_k \wedge i'_j > i'_k)$. This constraint ensures that the sequential order of features is respected[15], and that no feature may be matched twice. In the simplest form of Levenshtein[16], $sim^F(a, b) = 1$ if $a = b$ and 0 otherwise: for any $(v_{i_j}, v'_{i'_j}, s_j) \in E$, $s_j = sim^F(a_{i_j}, a'_{i'_j})$. Let

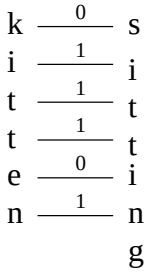
$$sim(G) = M - n_g \cdot cost_g - |E| + \sum_{(v_{i_j}, v'_{i'_j}, s_j) \in E} s_j,$$

where $M = \max(n, n')$ and n_g is the number of vertices that are not connected (i.e. the number of inserted or deleted words). $cost_g = 1$ in the simple Levenshtein form, but may be a parameter in the Needleman-Wunch variant (gap cost). In brief, the principle in this definition is to count the positions where no edit operation is needed: thus maximizing $sim(G)$ is equivalent to minimizing the cost of an alignment:

$align_{lev}(S, S') = G$, where G is any graph such that $sim(G) = \max(\{sim(G') | G' \in \mathcal{G}_{lev}\})$.

Finally, the function $score_{lev}$ is simply defined as $score_{lev}(G) = sim(G)/\max(n, n')$. It is not hard to see that this definition is equivalent to the usual one (see section 2): basically, the graph represents the concept called *trace* in (Wagner and Fischer, 1974), except that the cost function is "reversed" to become a similarity function.

Figure 1: Example of Levenshtein alignment



Suppose $cost_g = 1$:
$sim(G) = M - n_g - |E| + \sum_{e_j \in E} s_j$
$sim(G) = 7 - 1 - 6 + 4$
$sim(G) = 4$
$score_{lev}(G) = 4/7.$

3.2.2 Bag of features

For all simple measures using only sets of features, the function $align_{bag}(S, S')$ is defined in the following way: let $\mathcal{G}$ be the set of all graphs

[15] Constraints are a bit more complex for Damerau.

[16] In the Needleman-Wunch variant, sim^F should depend on the cost function, e.g.: $sim^F(a, b) = 1 - cost(a, b)$.

$G = (V, E)$ such that if $(v_{i_j}, v'_{i'_j}, s_j) \in E$ then $a_{i_j} = a'_{i'_j}$ (equivalently $sim^F(a_{i_j}, a'_{i'_j}) = 1$). Now let $once(\mathcal{G})$ be the set of all $G \in \mathcal{G}$ such that any pair of edges $(v_{i_j}, v'_{i'_j}, s_j), (v_{i_k}, v'_{i'_k}, s_k) \in E$ satisfies $i_j \neq i_k \wedge i'_j \neq i'_k$ (at most one match for each feature), and $a_{i_j} \neq a_{i_k}$ (a feature occurring several times is matched only once). Let $sim(G) = \sum_{(v_{i_j}, v'_{i'_j}, s_j) \in E} s_j$ for any $G = (V, E)$.

$align_{bag}(S, S') = G$, where G is any graph such that $sim(G) = \max(\{sim(G') \mid G' \in once(\mathcal{G})\})$.

Since all weights are equal to 1, one may show that $sim(G) = |S \cap S'|$ for any $G \in once(\mathcal{G})$. Thus the $score$ function is simply used for normalization, depending on the given measure: for example, $score_{overlap}(G) = \dfrac{sim(G)}{\min(n, n')}$.

3.2.3 Soft-TFIDF

The case of Cosine measure with TFIDF weighted vectors is a bit different. Here we define the SoftTFIDF version: let $align_{soft}(S, S')$ be the graph $G = (V, E)$ defined as[17] $(v_{i_j}, v'_{i'_j}, s_j) \in E$ if and only if $a'_{i'_j} = select(\text{CLOSEST}(\theta, a_{i_j}, S'))$, where CLOSEST is the function defined in section 2 and $select(E)$ is a function returning the first element in E if $|E| > 0$, and is undefined otherwise[18]. For any such edge, the weight s_j is

$$s_j = sim^F(a_{i_j}, a'_{i'_j}) \cdot \frac{\text{idf}(a_{i_j})}{n} \cdot \frac{\text{idf}(a'_{i'_j})}{n'}.$$

Once again, let $sim(G) = \sum_{(v_{i_j}, v'_{i'_j}, s_j) \in E} s_j$.

$score_{soft}(G) = sim(G)/(\|S\| \cdot \|S'\|)$, where

$$\|\langle a_1, \ldots, a_n \rangle\| = \sqrt{\sum_{i=1}^{n} \left(\frac{\text{idf}(a_i)}{n} \right)^2}.$$

Although it is not explicitly used in this definition, term frequency is taken into account through the number of edges: suppose a given term t appears m times in S and m' times in S', all m vertices corresponding to t in A (the partition representing S) will be connected to all m' vertices corresponding to t in A'. Thus there will be $m \times m'$ edges, which is exactly the unnormalized product of term frequencies $\text{tf}(t, S) \cdot \text{tf}(t, S') \cdot n \cdot n'$. Thus summing $m \times m'$ times $\text{idf}(t)/n \cdot \text{idf}(t)/n'$ in $sim(G)$ is equal to $\text{tfidf}(t, S) \cdot \text{tfidf}(t, S')$ (normalization is computed in the same way).

[17] In the simple case of CosTFIDF, the condition would be: $(v_{i_j}, v'_{i'_j}, s_j) \in E$ if and only if $a_{i_j} = a'_{i'_j}$. In other words, all identical features (and only they) are connected.

[18] "the first element" means that $select(E)$ may return any $e \in E$, provided the same element is always returned for the same set.

3.3 Meta-Levenshtein: Soft-TFIDF with Levenshtein alignment

We have shown in part 2.2 that there are some pitfalls in Soft-TFIDF, especially in the way the alignment is computed: no symmetry, possible score overflow. But experiments show that taking words IDF into account increases performance, and that Soft-TFIDF, i.e. the possible matching of words that are not strictly identical, increases performance (see section 4). That is why improving this kind of measure is interesting. Following the model we proposed above, we propose to mix the cosine-like similarity used in Soft-TFIDF with a Levenshtein-like alignment. The following measure, called *Meta-Levenshtein* (ML for short), takes IDFs into account but is not a bag-of-features metrics.

Let us define $align_{ML}$ in the following way: let $\mathcal{G}_{ML}$ be defined exactly as the set of graphs $\mathcal{G}_{lev}$ (see part 3.2.1), except that weights are defined as in the case of Soft-TFIDF: for any $G = (V, E) \in \mathcal{G}_{lev}$ and for any edge $(v_{i_j}, v'_{i'_j}, s_j) \in E$, let

$$s_j = sim^F(a_{i_j}, a'_{i'_j}) \cdot \frac{\text{idf}(a_{i_j})}{n} \cdot \frac{\text{idf}(a'_{i'_j})}{n'}.$$

Let $sim(G) = \sum_{(v_{i_j}, v'_{i'_j}, s_j) \in E} s_j$, and

$align_{ML}(S, S') = G$, where G is such that $sim(G) = \max(\{sim(G') \mid G' \in \mathcal{G}_{ML}\})$. Finally,

$$score_{ML}(G) = sim(G)/(\|S\| \cdot \|S'\|).$$

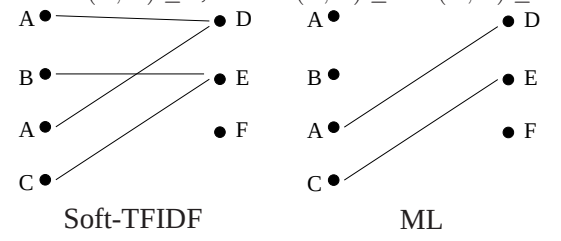
Compared to Soft-TFIDF, ML solves the problem of symmetry ($ML(S, S') = ML(S', S)$), and also the potential overflow, because no feature may be matched twice (see fig. 2). Of course, the alignment is less flexible in ML, since it must satisfy the sequential order of features. Practically, this measure may be efficiently implemented in the same way as Levenshtein similarity, including optionally the Damerau extension for transpositions. We have also tested a simple variant with possible extended transpositions, i.e. cases like ABC compared to CA, where both C and A are matched.

3.4 Recursive combinations for NE matching

One of the points we want to emphasize through the generic framework presented above is the mod-

Figure 2: Soft-TFIDF vs. ML alignment

With $sim(A, D) \geq \theta$, and $sim(C, E) \geq sim(B, E) \geq \theta$:



ularity of similarity measures. Our viewpoint is that traditional measures may be seen not only in their original context, but also as modular parameterized functions. The first application of such a definition is already in use in the form of measures like Monge-Elkan or Soft-TFIDF, which rely on some sub-measure to compare words inside NEs. But we will show that modularity is also useful at a lower level: measures concerning words may rely on similarity between (for example) n-grams, and even at this restricted level numerous possible kinds of similarity may be used.

Moreover, from the viewpoint of applications it is not very costly to compute similarities between n-grams and even between words. The number of n-grams is clearly bounded, and the number of words is not so high because there are only about 2 words by entity in average, and overall some words appear very often in entities[19].

4 Experiments

4.1 Data

Two corpora were used. Both contain mainly news and press articles, collected from various international sources. The first one, called "Iran Nuclear Threat" (INT in short), is in English and was extracted from the NTI (*Nuclear Threat Initiative*) web site[20]. It is 236,000 words long. Our second corpus, called "French Speaking Medias" (FSM in short), is 856,000 words long. It was extracted from a regular crawling of a set of French-speaking international newspapers web sites during a short time-frame (in July 2007). GATE[21] was used as the named entities recognizer for INT, whereas Arisem[22] performed the tagging of NEs for FSM. Recognition errors[23] appear in both corpora, but significantly less in FSM. We restricted the sets of NEs to those recognized as locations, organizations and persons, and decided to work only on entities appearing at least twice. Finally for INT (resp. FSM) we obtain 1,588 distinct NE (resp. 3,278) accounting altogether for 33,147 (resp. 23,725) occurrences.

Of course, it would be too costly to manually label as match (positive) or non-match (negative) the whole set containing $n \times (n-1)/2$ pairs, for the observed values of n. The approach consisting in labeling only a randomly chosen subset of pairs is ineffective, because of the disproportion between the number of negative and positive pairs (less than 0.1%). Therefore we tried to find all positive pairs, assuming the remaining lot are negative. Practically, the labeling step was based only on the best pairs as identified by a large set of measures[24]. The guidelines we used for labeling are the following: any incomplete, over-tagged or simply wrongly recognized NE is discarded. Then remaining pairs are classified as positive (coreferent), negative (non-coreferent), or "don't know"[25].

Corpus	Discarded	Pos.	Neg.	Don't know
INT	416 / 1,588	764	2,821	302
FSM	745 / 3,278	741	32,348	419

According to our initial hypotheses, all non-tagged pairs are considered as negative in the experiments below. "Don't know" pairs are ignored. As a further note, about 20% of the pairs are not orthographically similar (e.g. acronyms and their expansion): these pairs are out of reach of our techniques, and would require additional knowledge.

4.2 Observations

4.2.1 Taking IDF into account

To evaluate the contribution of IDF[26] in scoring the coreference degree between NE, let us ob-

[19] In the corpora we studied, 1172 NE (resp. 2533) contain 1107 distinct words (resp. 2785).

[20] `http://www.nti.org`

[21] `http://gate.ac.uk`

[22] `http://www.arisem.com`

[23] Mainly truncated entities, over-tagged entities, and common nouns beginning with a capital letter.

[24] This is a potential methodological bias, but we hope to have kept its effect as low as possible: the measures we used are quite diverse and do not assign good scores to the same pairs; therefore, for each measure, we expect that the potential misses (false negatives) will be matched by some other measure, thus allowing a fair evaluation of its performance. A few positive pairs are manually added (mainly acronyms).

[25] All ambiguous cases, mainly due to some missing precision (e.g. *"Ministry of Foreign Affairs"* and *"Russian Ministry of Foreign Affairs"*), and more rarely homonymy (e.g. *"Lebedev"* and *"*[*Valery*|*Oleg*] *Lebedev"*)

[26] It may be noticed that the Term Frequency in TFIDF is rarely important, since a given word appear almost always only once in a NE.

serve the differences among best scored pairs for measures Bag-of-words Cosine and Cosine over TFIDF weighted vectors. For example, the former will assign 0.5 to pair *"Prime Minister Tony Blair"/"Blair"* (from corpus INT), whereas the latter gives 0.61. As expected, IDF weights lighten the effect of non-informative words and strengthen important words. In both corpora, The F1-measure for TFIDF Cosine is about 10 points (in average) better than for Bag-of-words Cosine (see fig. 3).

4.2.2 Soft-TFIDF problems: normalization, threshold and sub-measure

As we have explained in section 2.2, the Soft-TFIDF measure (Cohen et al., 2003) may suffer from normalization problems. This is probably the reason why the authors seem to use it parsimoniously, i.e. only in the case words are very close (which is verified using a high threshold θ). Indeed, problems occur when the sub-measure and/or the threshold are not carefully chosen, causing performances drop: using Jaro measure with a very low threshold (0.2 here), performances are even worst than Bag-of-words cosine (see fig. 3). This is due to the *double matching* problem: for example, pair *"Tehran Times (Tehran)"/"Inter Press Service"* (from INT) is scored more than 1.0 because *"Tehran"* matches *"Inter"* twice: even with a low score as a coefficient, *"Inter"* has a high IDF compared to *"Press"* and *"Service"*, so counting it twice makes normalization wrong.

However, this problem may be solved by choosing a more adequate sub-measure: experiments show that using the CosTFIDF measure with bigrams or trigrams outperforms standard CosTFIDF. Of course, there are some positive pairs that are found "later" by Soft-TFIDF, since it may only increase score. But the "soft" comparison brings back to the top ranked pairs a lot of positive ones. In both corpora, the best sub-measure found is CosTFIDF with trigrams. *"Mohamed ElBaradei"/"Director Mohammad ElBaradei"* (INT) or *"Chine"/"China"* (FSM) are typical positive pairs found by this measure but not by standard CosTFIDF. Here no threshold is needed anymore because the sub-measure has been chosen with care, depending on the data, in order to avoid the normalization problem. This is clearly a drawback for Soft-TFIDF: it may perform well, but only with hand-tuning sub-measure and/or threshold.

4.2.3 Beyond Soft-TFIDF: (recursive) ML

In the FSM corpus, replacing Soft-TFIDF with (simple) Meta-Levenshtein at the word level does not decrease performance, even though the alignment is more constrained in the latter case. Using the same sub-measure to compare words (trigrams CosTFIDF), it does neither increase performance. A few positive pairs are missed in the INT corpus, due to the more flexible word order in English: *"U.S. State Department"/"US Department of State"* is such an example (12 among 764 are concerned). This problem is easily solved with the ML variant with extended transposition (see part 3.3): in both corpora, there are no positive pairs requiring more than a gap of one word in the alignment. Thus this measure is not only performant but also robust, since it does not need any hand-tuning.

As a second step, we want to improve results by selecting a more fine-grained sub-measure. We have tried several ideas, such as using different kinds of n-grams similarity inside the words similarity measure. Firstly, trigrams performed better than bigrams or simple characters. Secondly, the best trigrams similarity method found is actually very simple: it consists in using CosTFIDF computed on the *trigrams contexts*, i.e. the set of closest[27] trigrams of all occurrences of the given trigram. Unsurprisingly, good scores are generally obtained for pairs of trigrams that have common characters. But it seems that this approach also enhances robustness, because it finds similarities between "close characters": in the French corpus, one observes quite good scores between trigrams containing an accentuated version and the non accentuated version of the same character. Furthermore, some character encoding errors are somehow corrected this way[28]. This is possibly the reason why the improvement of results is better in FSM than in INT (see table 1).

Finally, using also ML to compute similarity *between* words[29] yields the best results. This means that compared to the simple CosTFIDF submeasure, one does not compare bags of trigrams but ordered sequences of trigrams[30].

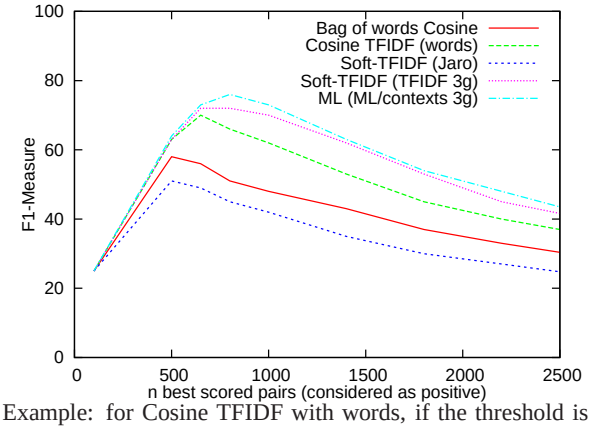
[27] We have tried different window sizes for such contexts, from 2 to 10 trigrams long: performances were approximately the same. We only consider trigrams found in the entities.

[28] For example, the *ï* in the name *"Lugovoï"* appears also in FSM as *i*, as *y*, as *à*, and is sometimes deleted.

[29] i.e. not only between sequences of words: in this case ML is run between trigrams at the word level, and then another time between words at the NE level.

[30] It is hard to tell whether it is the sequential alignment or

Figure 3: F1-Measures for FSM (percentages)



Example: for Cosine TFIDF with words, if the threshold is set in such a way that (only) the 1000 top ranked pairs are classified as positive, then the F1-measure is around 60%.

Table 1: Best F1-measures (percentages)

	INT			FSM		
Measure	F1	P	R	F1	P	R
Cosine	51.6	63.2	43.6	59.5	76.2	48.7
CosTFIDF	62.6	71.7	55.6	69.9	84.2	59.8
Soft TFIDF/3g	68.6	74.2	63.9	73.1	79.8	67.6
ML/ML-context	70.6	72.6	68.7	77.0	82.5	72.2

P/R: Corresponding Precision/Recall.

4.3 Global results

Results are synthesized in table 1, which is based on the maximum F1-measure for each measure. One observes that F1-measure is 3 to 6 points better for Soft-TFIDF than for standard TF-IDF, and that our measure still increases F1-measure by 2 (INT) to 4 points (FSM). Results show that its contribution consists mainly in improving the recall, which means that our measure is able to catch more positive pairs than Soft-TFIDF: for example, the pair *"Fatah Al Islam"/ "Fateh el-Islam"* (FSM) is scored 0.54 by SoftTFIDF and 0.70 by ML. Our measure remains the best for all values of n in fig. 3, and results are similar for F0.5-measure and F2-measure: thus, irrespective of specific application needs which may favor precision or recall, ML seems preferable.

5 Conclusion

In conclusion, we have proposed a generic model to show that similarity measures may be combined in numerous ways. We have tested such a combination, based on Soft-TFIDF, which performs better than all existing similarity metrics on two corpora. Our measure is robust, since it does not rely on any kind of prior knowledge. Thus it may be easily used, in particular in applications where NE matching is useful but is not the essential task.

the "right" use of the trigrams sub-measure which is responsible for the improvement, since the only possible comparison at this level is Soft-TFIDF.

Acknowledgements

This work has been funded by the National Project Cap Digital - Infom@gic. We thank Loïs Rigouste (Pertimm) and Nicolas Dessaigne and Aurélie Migeotte (Arisem) for providing us with the annotated French corpus.

References

Bilenko, Mikhail, Raymond J. Mooney, William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23.

Christen, Peter. 2006. A comparison of personal name matching: Techniques and practical issues. Technical Report TR-CS-06-02, Department of Computer Science, The Australian National University, Canberra 0200 ACT, Australia, September.

Cohen, William W., Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In Kambhampati, Subbarao and Craig A. Knoblock, editors, *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 9-10, 2003, Acapulco, Mexico*, pages 73–78.

Freeman, Andrew, Sherri L. Condon, and Christopher Ackerman. 2006. Cross linguistic name matching in English and Arabic. In Moore, Robert C., Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *Proc. HLT-NAACL*.

Navarro, Gonzalo. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Pouliquen, Bruno, Ralf Steinberger, Camelia Ignat, Irina Temnikova, Anna Widiger, Wajdi Zaghouani, and Jan Zizka. 2006. Multilingual person name recognition and transliteration. *CORELA - Cognition, Representation, Langage.*

Prager, John, Sarah Luger, and Jennifer Chu-Carroll. 2007. Type nanotheories: a framework for term comparison. In *Proceedings of CIKM '07*, pages 701–710, New York, NY, USA. ACM.

Wagner, R. and M. Fischer. 1974. The string-to-string correction problem. *JACM*, 21(1):168–173.

Winkler, W. E. 1999. The state of record linkage and current research problems. Technical Report RR99/04, US Bureau of the Census.