A Flexible Measure of Contextual Similarity for Biomedical Terms

I. Spasic and S. Ananiadou

Pacific Symposium on Biocomputing 10:197-208(2005)

A FLEXIBLE MEASURE OF CONTEXTUAL SIMILARITY FOR BIOMEDICAL TERMS*

I. SPASIĆ[1], S. ANANIADOU[2]

[1]*Department of Chemistry, UMIST, Manchester, UK*
[2]*School of Computing, Science and Engineering, University of Salford, UK*
E-mail: i.spasic@umist.ac.uk, s.ananiadou@salford.ac.uk

We present a measure of contextual similarity for biomedical terms. The contextual features need to be explored, because newly coined terms are not explicitly described and efficiently stored in biomedical ontologies and their inner features (e.g. morphologic or orthographic) do not always provide sufficient information about the properties of the underlying concepts. The context of each term can be represented as a sequence of syntactic elements annotated with biomedical information retrieved from an ontology. The sequences of contextual elements may be matched approximately by edit distance defined as the minimal cost incurred by the changes (including insertion, deletion and replacement) needed to transform one sequence into the other. Our approach augments the traditional concept of edit distance by elements of linguistic and biomedical knowledge, which together provide flexible selection of contextual features and their comparison.

1. Introduction

Breakthrough advances in biotechnology have given rise to rapid production of biomedical data. New discoveries are being described in scientific papers (most often electronically available) with the intention of sharing the results with the scientific community. However, the rapid expansion of the bioliterature[a] makes it increasingly difficult to locate the right information at the right time. Clearly, for biomedical experts to experience the full benefits of electronically accessible literature, natural language processing (NLP) applications (such as information retrieval, information extraction,

*This work has been partially supported by the JISC-funded National Centre for Text Mining (NaCTeM), Manchester, UK.

[a]For example, the MEDLINE database (www.ncbi.nlm.nih.gov/PubMed) currently contains approximately 12 million references to biomedical articles, growing by more than 10,000 references weekly.

etc.) are becoming a necessity in order to facilitate navigation through huge volumes of biomedical texts.

Automatic extraction and retrieval of biomedical information subsumes identification of terms denoting biomedical concepts (such as compounds, genes, drugs, reactions, etc.), their properties and mutual relations from a corpus of relevant documents. Rule-based approaches to these problems cannot cope with an enormous and ever growing number of terms and the complex structure of terminologies[b]. Since rules would need to be defined for each NLP task and biomedical subdomain separately, manual rule engineering in such a broad and complex domain is hindered by inefficiency and inconsistency. Alternatively, a similarity measure could be used as a vehicle of machine learning approaches to a variety of NLP tasks, utilising the large body of biomedical texts as the training data. In this paper, we suggest a measure of contextual similarity between biomedical terms based on edit distance. The alignment between two contexts corresponding to their edit distance can be used to match terms occurring in similar contexts. This property can be further exploited to support tasks such as term classification and disambiguation, extraction of their relations, etc.

The remainder of the paper is organised as follows. In Section 2 we briefly overview edit distance. Section 3 introduces the SOLD measure, generally based on the idea of edit distance as a means of assessing contextual similarity of biomedical terms. Sections 4, 5 and 6 give details on the specific aspects of the SOLD measure, namely syntactic, ontology-driven and lexical components. Finally, in Section 7 we conclude the paper.

2. Background and Related Work

Edit distance (ED) has been widely used for approximate string matching, where the distance between identical strings equals zero and increases as the strings get more dissimilar with respect to the symbols they contain and the order in which they appear. ED is defined as the minimal cost incurred by the changes needed to transform one string into the other. These changes may include insertion or deletion of a single character, replacement of two characters in the two strings and transposition of two adjacent characters in a single string. The choice of edit operations and their costs influences the "meaning" of the corresponding approximate matching, and thus depends

[b]For example, UMLS (www.nlm.nih.gov/research/umls) currently contains over one million concepts named by 2.8 million terms, organised into a hierarchy of 135 classes and interconnected by 54 different relations.

on a specific application.

A most popular application area of ED is molecular biology, where it has been used to compare DNA and protein sequences in order to infer information about the common ancestry, functional equivalence, possible mutations, etc.[1] It has also been successfully utilised in NLP to deal with alternate spellings, misspellings, the use of upper- and lower-case letters, etc. Further, ED has been used in terminological processing for the recognition of term variants (namely, protein names) based on their *internal* properties focusing on orthographic features.[2] Our intention, however, is primarily to explore *contextual* properties of terms.

In this case, it is more convenient to apply ED at the *word level* rather than the *character level*. Namely, character-based ED does not cope well with permutations of words. For instance, judging by the "conventional" ED, *stone in kidney* is more similar to *stone in bladder* than *kidney stone*. Obviously, for some applications it is more useful to treat strings as sequences of words. For example, approximate string matching can be viewed as the problem of pairing up their words so as to minimise their ED.[3] Recently, ED has been applied at the word level[4] as support for extended phrase-based text search allowing different wordings and syntactic mistakes. In this approach, ED was simply applied to words as opposed to characters. We, however, enriched the basic ED approach with linguistic knowledge (relying on part-of-speech (POS) tagging and partial parsing) and domain-specific knowledge (using an ontology). In the following section, we point to the main developments in this direction.

3. Approximate Context Matching

ED usually relies on the exact matches between symbols unless "wild card" symbols are allowed. This is not suitable for words, because they are inflected. Also, term variation causes two terms not to match even when they are synonymous. We want to keep the main idea of ED to account for different orderings of words, but also to make it more flexible towards lexical variations. For example, two inflected word forms should match if both their lexical categories and their base forms are identical, e.g.: [c]

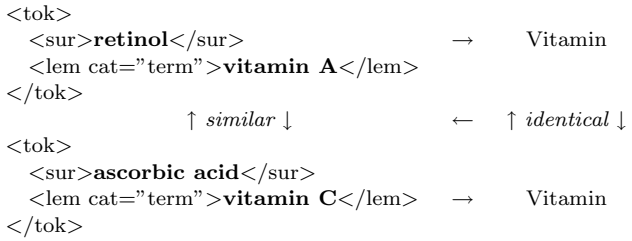
<tok><sur>**better**</sur><lem cat="**adj**">**good**</lem></tok>
<tok><sur>**good**</sur> <lem cat="**adj**">**good**</lem></tok>

[c] In the given XML notation, elements `<tok>`, `<sur>` and `<lem>` stand for token, surface form and lemma respectively, while attribute `cat` corresponds to category.

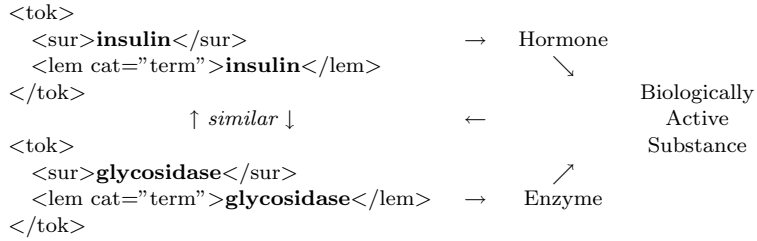
When two terms are compared, information from an ontology may be utilised. If the terms match exactly or if they are identified as variants, the matching score should be the highest, slightly less if they are siblings in the "is-a" hierarchy, etc. For example, the following terms have been recognised as variants in UMLS and annotated as such in the corpus by mapping them to the same preferred term form:

<tok><sur>**vitamin A**</sur><lem cat="term">**vitamin A**</lem></tok>
<tok><sur>**A vitamin**</sur><lem cat="term">**vitamin A**</lem></tok>
<tok><sur>**vitamin-A**</sur><lem cat="term">**vitamin A**</lem></tok>
<tok><sur>**retinol**</sur> <lem cat="term">**vitamin A**</lem></tok>

Similarly, classified terms can be compared through their classes, e.g. both *retinol* and *ascorbic acid* are mapped to the *Vitamin* class in UMLS, and therefore can be regarded similar:



When term classes are not identical, their superclasses can be compared analogously, e.g. the classes retrieved for terms *insulin* and *glycosidase* are respectively *Hormone* and *Enzyme*, which both descend from the *Biologically Active Substance* class, so the given terms can be regarded similar:



When at least one of the compared terms is not classified, then lexical clues may indicate their semantic similarity. For example, suppose that *retinol* has been mapped to its preferred form *vitamin A* in the ontology and that *vitamin C* has only been identified in the corpus. Then the lexical similarity between the corresponding normalised forms *vitamin A* and *vitamin C* (e.g. measured by ED) can be used to reduce the cost of their replacement:

<tok>
  <sur>**retinol**</sur> → Vitamin
  <lem cat="term">**vitamin A**</lem>
</tok>
↑ *similar* ↓ ← ↑ *lex. similar* ↓ ← ↑ ? ↓
<tok>
  <sur>**vitamin C**</sur>
  <lem cat="term">**vitamin C**</lem> → ?
</tok>

Apart from lexical and terminological clues, syntactic information can be utilised as well. For example, partial parsing can be applied to POS-tagged text to group subsequent words into basic syntactic structures (e.g. noun phrases). ED applied to chunks of words rather than individual words is "forced" to take into account the syntactic structure at the phrase level. By choosing to replace syntactic categories with similar properties at lower costs (e.g. nouns and pronouns), ED can also be used to compare the syntactic structure at the sentence level. Namely, the sentences receiving low ED values are the ones that can be transformed into one another using a small number of low-cost edit operations, implying that their overall syntactic structure is fairly isomorphic.

We suggested how the traditional concept of ED can be augmented by elements of linguistic and domain-specific knowledge. We continue to describe the specific solutions used to implement the SOLD (syntactic, ontology-driven, lexical distance) measure.

4. Syntactic Component

Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ denote two sentences as the sequences of chunks (not individual words) denoted by x_i $(1 \leq i \leq m)$ and y_j $(1 \leq j \leq n)$. Their distance is defined as the minimal number of edit operations necessary to transform X into Y. Figure 1 describes the computation of the SOLD measure using the standard dynamic programming approach,[5] where $\text{cost}(i, j)$ denotes ED between $(x_1, \ldots, x_i)$ and $(y_1, \ldots, y_j)$, IC and DC (where IC $\equiv$ DC) are the costs of inserting and deleting a given chunk, and RC is the cost of replacing two chunks (see Table 1[d]). Automatic optimisation of the cost function led to overfitting. Therefore, an appropriate cost function has been chosen empirically and supported by experiments conducted with equal weights, metric and non-metric cost functions. We

[d] The cost of replacement by the epsilon symbol represents the cost of inserting or deleting the other symbol.

describe the motivation behind the chosen cost function, which provided satisfactory results.

$$
\begin{array}{l}
\text{cost}(0,0) = 0; \\
\text{for } (i = 1;\ i \leq m;\ i = i + 1) \quad \text{cost}(i,0) = \text{cost}(i-1,0) + \text{IC}(x_i); \\
\text{for } (j = 1;\ j \leq n;\ j = j + 1) \quad \text{cost}(0,j) = \text{cost}(0,j-1) + \text{DC}(y_j); \\
\text{for } (i = 1;\ i \leq m;\ i = i + 1) \\
\text{for } (j = 1;\ j \leq n;\ j = j + 1) \\
\qquad \text{cost}(i,j) = \min \left\{ \begin{array}{ll} \text{cost}(i-1,j) & + \text{IC}(x_i) \\ \text{cost}(i,j-1) & + \text{DC}(y_j) \\ \text{cost}(i-1,j-1) & + \text{RC}(x_i, y_j) \end{array} \right\}; \\
\text{sold}(X,Y) = \text{cost}(m,n);
\end{array}
$$

Figure 1. Calculation of the SOLD distance.

The choice of specific costs is based on an assumption about the potential semantic content and importance of syntactic chunks. Deleting a term incurs the highest possible cost (1), since important domain-specific information is lost. High importance (0.9) is also given to verbs as they may represent domain-specific relations. Generally, noun phrases (NPs), together with verbs, carry "heavy" semantic load. This is emphasised even more in a sublanguage, because terms constitute a subclass of NPs. NPs other than terms are still semantically important, especially since they can be unrecognised terms, so they are assigned high cost (0.9). Further, pronouns are given high cost (0.85), because they can co-refer with terms (i.e. indirectly denote a domain-specific concept). Prepositions can model spatial, temporal and other types of relationships between terms, and for this reason they are relatively highly ranked (0.5). Similarly, adjectives and adverbs as potential modifiers of terms and domain-specific verbs, e.g.:

> *... the fragment of SMRT encoding amino acids 1192-1495, which* ***strongly*** *interacts with TRbeta,* *interacts* ***very weakly*** *with COUP-TFI ...*

are given the same cost (0.5). Next ranked (0.4) are different forms of the verb *to be*, which can be used in a general sense, but can also model the "is-a" relationship between terms, e.g.:

> *... acetylcysteine* ***is a*** *drug usually used to reduce the thickness of mucus ...*

so they are assigned a similar cost (0.4). Auxiliary verb phrases can be used to modify the meaning of domain-specific verbs and in that manner encode important semantic information, e.g.:

> *... the oestrogen receptor AF-2 antagonist hydroxytamoxifen* ***cannot*** *promote ER-TIF1 interaction ...*

and they incur the same cost (0.4). A low cost (0.2) is given to conjunctions, whose main role is to support text cohesion and not to pass relevant domain-specific information. Other chunks are assigned zero cost, as they are regarded irrelevant. For example, linking phrases (e.g. *on the other hand*) guide a reader, but carry no explicit semantic content. Punctuation marks are used similarly to improve the readability, and are thus discarded. Determiners are ignored because of their insufficient semantic content and especially because they are not used consistently or even correctly.

Table 1. The cost of edit operations for different chunks.

Chunk	np	term	link	be	aux	adj	adv	cnj	det	prep	pron	pun	v	ϵ
np	0.20	0.30	0.90	0.90	0.90	0.75	0.90	1.00	0.90	1.00	0.15	0.90	0.70	0.90
term	0.30	0.15	1.00	0.90	0.90	0.80	0.95	1.00	1.00	1.00	0.15	1.00	0.70	1.00
link	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
be	0.90	0.90	0.40	0.00	0.10	0.90	0.75	0.55	0.40	0.70	0.90	0.40	0.55	0.40
aux	0.90	0.90	0.40	0.10	0.00	0.90	0.75	0.55	0.40	0.70	0.90	0.40	0.55	0.40
adj	0.75	0.80	0.50	0.90	0.90	0.15	0.25	0.65	0.50	0.85	0.75	0.50	0.90	0.50
adv	0.90	0.95	0.50	0.75	0.75	0.25	0.15	0.65	0.50	0.85	0.90	0.50	0.80	0.50
cnj	1.00	1.00	0.20	0.55	0.55	0.65	0.65	0.00	0.20	0.55	1.00	0.20	0.95	0.20
det	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
prep	1.00	1.00	0.50	0.70	0.70	0.85	0.85	0.55	0.50	0.05	1.00	0.50	1.00	0.50
pron	0.15	0.15	0.85	0.90	0.90	0.75	0.90	1.00	0.85	1.00	0.05	0.85	0.70	0.90
pun	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
v	0.70	0.70	0.90	0.55	0.55	0.90	0.80	0.95	0.90	1.00	0.70	0.90	0.20	0.90
ϵ	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.90	0.00	0.90	

The costs of replacing two chunks depends on their types. Zero cost is used to make the chunks fully compatible (e.g. auxiliary verb phrases can freely interchange). Also, all chunks that are deleted with zero cost may freely replace one another. Generally, the replacement costs reflect the compatibility between the involved chunks. Therefore, low costs can be found along the main diagonal in Table 1 emphasising the highest compatibility between the same chunk types. An exception with this regard is the cost of replacing NPs and terms with pronouns (0.15), which can act as "wild cards" for these chunks. Note that the cost of replacing the same chunk types is not necessarily zero. Although compatible, they cannot always be freely replaced. This is used for high-content chunks (such as terms and NPs) in order to emphasise the importance of semantic information they encode and not only their syntactic function.

So far we have mostly relied on syntactic information acquired through POS tagging and partial parsing. We would like to incorporate more domain-specific knowledge into the SOLD measure in order to support se-

mantic comparison. Since the ontology used incorporates hierarchies of terms and domain-specific verbs, the replacement costs can be fine-grained so as to reflect the semantic closeness of terms and verbs considered. The actual replacement cost involving such chunks depends on their content. In these cases, the replacement cost r given in Table 1 is not fixed, but rather represents its upper limit. There are two basic principles used for the calculation of the replacement cost in such cases: a knowledge-rich approach based on domain-specific knowledge contained in the ontology and a knowledge-poor approach based on lexical similarity. In the following sections we discuss these two approaches to semantic comparison.

5. Ontology-Driven Component

Ontology-based replacement cost is used for terms and verbs contained in the ontology. Let us describe how the replacement cost is calculated for two classified terms. Figure 2 describes the computation of the replacement cost (RC) for two classified terms (t_1 and t_2) based on their similarity: the higher the similarity, the smaller the replacement cost. It is first checked if the terms are lexical variants, that is – if they are orthographic variants (differing in the use of hyphenation, lower and upper cases, spelling, etc.) or inflectional variants (differing in number – singular or plural), simply by checking if they are linked to the same term identifier in UMLS. Lexical variants are given the highest similarity value (1), since they identify the same concept and differ only in their textual realisation. For the same reason, semantic variants (i.e. synonyms) are given the same similarity score. It is checked if two terms are synonyms, by checking if they are mapped to the same concept identifier in UMLS. If they are not recognised as semantic variants, the class information is used for their further comparison. All semantic classes in UMLS are organised into a hierarchy, which can be used to quantify their similarity. The tree similarity (ts) between two classes (C_1 and C_2) is calculated according to the following formula:

$$\mathrm{ts}(C_1, C_2) = \frac{2 \cdot \mathrm{common}(C_1, C_2)}{\mathrm{depth}(C_1) + \mathrm{depth}(C_2)}$$

where $\mathrm{common}(C_1, C_2)$ denotes the number of common classes in the paths leading from the root to the given classes, and $\mathrm{depth}(C)$ is the number of classes in the path connecting the root and the given class. This formula is a derivative of Dice coefficient, where each ancestor class is treated as a separate feature. It was previously used to measure conceptual similarity in a hierarchically structured lexicon.[6] Other measures have been proposed

and could be used as well. For example, Resnik[7] used a "probabilistic variation" of this model:

$$\mathrm{ts}(C_1, C_2) = \frac{2 \cdot \log P(S(C_1, C_2))}{\log P(C_1) + \log P(C_2)}$$

where $S(C_1, C_2)$ is the deepest class that subsumes both classes, and $P(C)$ denotes the probability that a random object belongs to the given class. To be used with UMLS, this approach would require additional computation of these probabilities.

Further, since UMLS supports multiple classification, we estimate the similarity between two terms as the maximal similarity between their classes. Note that if two terms belong to the same class (among others if any), then their similarity reaches 1. In order to differentiate between compatible terms (i.e. non-synonymous terms belonging to the same class) and semantic variants (i.e. synonymous terms), we scale down the tree similarity by 10%. Finally, having calculated the similarity between two terms, it is converted to the corresponding distance and mapped to the interval $[0, r]$ (where $r = 0.15$ is the maximal replacement cost for two terms). The replacement cost for classified domain-specific verbs is calculated similarly.

if t_1 and t_2 are lexical variants,	then	$sim(t_1, t_2) = 1$,
else if t_1 and t_2 are synonyms,	then	$sim(t_1, t_2) = 1$,
else		$sim(t_1, t_2) = 0.9 \cdot \max$,
		(where max is the maximal value of $\mathrm{ts}(C_1, C_2)$ for all classes C_1 of t_1 and all classes C_2 of t_2)
$\mathrm{RC}(t_1, t_2) = r \cdot (1 - sim(t_1, t_2))$;		

Figure 2. Calculation of the variable replacement costs.

6. Lexical Component

The approach described in Section 5 applies only to *classified* terms or verbs. Currently, biomedical ontologies are inherently incomplete due to the fast-growing number of terms. Therefore, it would be useful to use clues other than the ones explicitly stated in the ontology in order to extend the semantic comparison to other terms and verbs. Since the syntactic clues are already being used when comparing term contexts by the SOLD distance, we opted for internal lexical properties of context constituents. Lexical comparison has been enabled for terms, NPs and verbs. We utilised the standard ED approach applied at the grapheme level. Similarly to Tsuruoka and Tsujii,[2] we differentiate between four types of graphemes (space and hyphen, digits, letters and all other graphemes) in order to determine the appropriate costs of edit operations (Table 2).

Table 2. The cost of edit operations for different graphemes.

Grapheme	" " or "–"	digit	letter	other	ϵ
" " or "–"	0.05	1.00	1.00	1.00	0.50
digit	1.00	0.10	1.00	1.00	1.00
letter	1.00	1.00	0.90*	1.00	1.00
other	1.00	1.00	1.00	0.05	0.50
ϵ	0.50	1.00	1.00	0.50	

The highest deletion costs are given to digits and letters as they convey more information compared to other graphemes. For example, spaces and hyphens basically serve to improve the readability of multi-word terms. In addition, they are not always used consistently and often cause orthographic variation by replacing each other or being omitted altogether (e.g. *EGR-1* vs. *EGR 1* vs. *EGR1*). Hence, these graphemes are assigned lower cost (0.5). The same cost is given to all other graphemes for similar reasons.

The replacement cost is generally chosen so as to "discourage" the replacement of graphemes of different types (e.g. digits and letters) by assigning the highest cost (1) to such operations. The replacement within the same type depends on the importance and similarity between the graphemes. Space and hyphen are regarded similar, thus are given a low cost (0.05). Similarly, digits can be interchanged at a relatively low cost (0.1). Letters are given high cost (0.9) since morphemes (as groups of letters), often in the form of neoclassical roots and affixes,[8] are used to encode important features of biomedical concepts. In order to make the cost function less case-sensitive, the cost of replacing the same letter differing only in case is obtained by subtracting the general replacement cost for letters from the highest possible cost: $1-0.9 = 0.1$. Note that we still maintain the case sensitivity. This may be important for some types of terms (e.g. case variants sometimes can be used to distinguish a gene from its protein[9]). ED between two chunks is used to adjust the cost of their replacement.

The lexical component adds to the robustness of the SOLD measure by comparing terms and verbs not covered by the ontology and, therefore, overlooked by the ontology-driven component. It also makes the SOLD distance approach robust with respect to spelling variations and typing errors occurring within semantically important chunks. Alternatively, word edit distance[3] or the MetaMap[10] program for the recognition of term variants can be used to support lexical comparison.

7. Discussion and Conclusions

We described the SOLD measure, which can be used to assess similarity between terms based on their contextual features. Compared to other mea-

sures of contextual similarity, our approach offers a significant degree of flexibility. For example, Nenadić et al.[11] relied on Dice coefficient using predefined lexico-syntactic patterns as contextual features. Such an approach lacks the necessary flexibility, since small syntactic variations may cause similar patterns not to match and to be accounted as different features. In our approach, the variability of a natural language has been accounted for at multiple levels. First, the choice of contextual features need not be rigidly predefined, as features may be matched, replaced or discarded as necessary through elastic matching, thus neutralising some types of syntactic variation. Further, lexical variability is partly neutralised by using an ontology to match different forms of terms and domain-specific verbs. In addition, lexical similarity is assessed by ED in order to compensate for incompleteness of the ontology.

We also compare our approach to that of Dagan et al.[12], who proposed a method for estimating word similarity from sparse data, the main assumption being that similar word co-occurrences should have similar mutual information. In our approach, sparsity of data can be partly compensated by non-exact matching driven by the ontology and lexical similarity. In other words, classes of terms (both lexical and semantic) are compared rather than individual terms, which means that individual frequencies are aggregated into collective frequencies of similar terms. In addition, syntactic knowledge and ED are used to generalise a rigid and knowledge-poor notion of co-occurrence into contextual similarity that takes into account not only the relative position and the frequency of co-occurrence, but also a wider context with respect to its syntactic structure and semantic content.

Word similarity approaches can be evaluated through the recognition of synonyms.[13] We generalise this idea to recognition of terms belonging to the same semantic classes in order to evaluate the SOLD measure. It has been fully implemented as part of a case-based reasoning system in which the similarity measure plays a key inferencing role.[14] The efficiency of comparison is improved through retrieval component developed to reduce the search space to potentially similar cases with respect to the SOLD measure, thus avoiding a brute-force nearest-neighbour approach and enhancing the scalability of the given measure. We tested our approach for functional classification of chemicals (13 UMLS classes) based on a training corpus of 2072 MEDLINE abstracts annotated with 18236 training, 2405 validation, 2838 testing and 30419 non-classified terms. The performance has been evaluated relative to three baseline methods (random, naive Bayes and rule-based classifier), which achieved 8.97%, 25.31% and 45.54% for

F-measure, while we obtained 58.52%.

We plan to use the SOLD measure to improve the flexibility of a rule-based information extraction (IE) system by identifying contexts similar to the ones to which the IE rules apply directly and to extract information of interest indirectly by the rules through alignment. In particular, we are interested in extracting information on protein-protein interactions. We believe that other methods developed for NLP tasks in biomedicine may similarly be facilitated by the use of the SOLD measure.

References

1. A. Apostolico and R. Giancarlo. Sequence Alignment in Molecular Biology. In M. Farach-Colton et al. (Eds.), *DIMACS Special Year for Mathematical Support of Molecular Biology*, 47:85-116, 1999.
2. Y. Tsuruoka and J. Tsujii. Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. In *ACL Workshop on NLP in Biomedicine*, Sapporo, Japan, 41-48, 2003.
3. J. French, A. Powell and E. Schulman. Applications of Approximate Word Matching in Information Retrieval. In *Int Conf on Knowledge and Information Management*, Los Angeles, USA, 1997.
4. G. Navarro, E. Silva de Moura, M. Neubert, N. Ziviani and R. Baeza-Yates. Adding Compression to Block Addressing Inverted Indexes. *Information Retrieval*, 3:49-77, 2000.
5. R. Wagner and M. Fischer. The String-to-String Correction Problem. *J of ACM*, 21(1):168-173, 1974.
6. Z. Wu, M. Stone Palmer. Verb Semantics and Lexical Selection. In *Annual Meeting of the ACL*, Las Cruces, USA, 133-138, 1994.
7. P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Int Conf on Artificial Intelligence*, Montreal, Canada, 448-453, 1995.
8. S. Ananiadou. A Methodology for Automatic Term Recognition. In *COLING*, Kyoto, Japan, 1034-1038, 1994.
9. M. Weeber, B. Schijvenaars, E. van Mulligen, B. Mons, R. Jelier, C. van der Eijk and J. Kors. Ambiguity of Human Gene Symbols in LocusLink and MEDLINE: Creating an Inventory and a Disambiguation Test Collection. In *AMIA Symposium*, 704-708, 2003.
10. A. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: the MetaMap Program. In *AMIA Symposium*, 17-21, 2001.
11. G. Nenadić, I. Spasić and S. Ananiadou. Mining Term Similarities from Corpora. *Terminology*, 10(1):55-80, 2004.
12. I. Dagan, S. Marcus and S. Markovitch. Contextual Word Similarity and Estimation from Sparse Data. *Computer, Speech and Language*, 9:123-152, 1995.
13. G. Grefenstette. *Exploration in Automatic Thesaurus Discovery*. 1994.
14. I. Spasić. *A Machine Learning Approach to Term Classification*. PhD Thesis. University of Salford, UK, 2004.