IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 12, 2018, accepted April 9, 2018, date of publication April 12, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2826224

Head to Head: Semantic Similarity of Multi–Word Terms

IRENA SPASIĆ[1], PADRAIG CORCORAN[1], ANDREI GAGARIN[2], AND ANDREAS BUERKI[3]

[1]School of Computer Science and Informatics, Cardiff University, Cardiff, CF10 4PE, U.K.
[2]School of Mathematics, Cardiff University, Cardiff, CF24 4AG, U.K.
[3]School of English, Communication and Philosophy, Cardiff University, Cardiff, CF10 3EU, U.K.

Corresponding author: Irena Spasić (spasici@cardiff.ac.uk)

ABSTRACT Terms are linguistic signifiers of domain–specific concepts. Semantic similarity between terms refers to the corresponding distance in the conceptual space. In this paper, we use lexico-syntactic information to define a vector space representation in which cosine similarity closely approximates semantic similarity between the corresponding terms. Given a multi–word term, each word is weighed in terms of its defining properties. In this context, the head noun is given the highest weight. Other words are weighed depending on their relations to the head noun. We formalized the problem as that of determining a topological ordering of a direct acyclic graph, which is based on constituency and dependency relations within a noun phrase. To counteract the errors associated with automatically inferred constituency and dependency relations, we implemented a heuristic approach to approximating the topological ordering. Different weights are assigned to different words based on their positions. Clustering experiments performed on such a vector space representation showed considerable improvement over the conventional bag–of–word representation. Specifically, it more consistently reflected semantic similarity between the terms. This was established by analyzing the differences between automatically generated dendrograms and manually constructed taxonomies. In conclusion, our method can be used to semi–automate taxonomy construction.

INDEX TERMS Semantic similarity, natural language processing, clustering methods, knowledge acquisition.

I. INTRODUCTION

A term is intuitively defined as a noun phrase that occurs frequently in a domain–specific discourse and has a special meaning in the given domain [1], [2]. In other words, terms are linguistic signifiers of domain–specific concepts [3]. As such, they are basic means of conveying scientific and technical information [4]. In comparison to other words and phrases used in a sublanguage, terms carry heavier information load. It is, therefore, essential to build and maintain terminologies in order to enhance the performance of many natural language processing (NLP) applications.

All terms belonging to a specific domain collectively form its terminology [3]. Bodenreider *et al.* [5] emphasize the structured nature of a terminology with the hierarchy being the main organizational principle. Most terminologies use hierarchies based on a relation of dominance that comprises the taxonomic (is–a) relation and the meronymic (part of) relation with the former used most commonly in practice. This implies that a terminology is not merely a collection of terms, but rather a structure imposed over such collection.

The relations between concepts can be mapped to lexical relations between the corresponding terms [6]. Lexical semantics defines four types of congruence relations: identity, inclusion, overlap and disjunction [7]. In many cases, such relations between terms can be inferred by simply comparing their bag–of–words (BOW) representations. For example, two terms $t_1 =$ *effective contraceptive method* and $t_2 =$ *effective method of contraception*can be mapped to the same BOW representation, $\mathrm{BOW}(t_1) = \mathrm{BOW}(t_2) = \{$*effect*, *contracept*, *method*$\}$, where the stop words have been removed and the remaining content stemmed. This type of reasoning is used in FlexiTerm [8], an automatic term recognition (ATR) system, to infer that the given terms refer to the same domain–specific concept, i.e. that they are synonyms. Synonymy is the lexical relation that corresponds to identity. In the context of terminology structuring, an equally important lexical relation is that of hyponymy [9], [10]. It corresponds to inclusion, which

we mentioned previously as the main vehicle for adding taxonomic structure to terminologies. Most studies on extracting hyponymy relations from text focus on external context of the participating terms using lexico–syntactic patterns and/or distributional semantics, e.g. [11]–[17]. In this study, we are focusing on multi–word terms (MWTs) and relations between them that can be inferred from their content. In some cases, the BOW approach may be used to identify hyponymy relation between terms. For example, the subsumption relationship between the BOW representatives of two terms t_1 = *anterior cruciate ligament* and t_2 = *cruciate ligament* can be used to infer that t_1 is a hyponym of t_2. However, the BOW approach is insufficient (e.g. *complete tear of anterior cruciate ligament* is not a hyponym of *anterior cruciate ligament*), because identification of hyponymy very much depends on the analysis of syntactic relations, mainly the head–modifier relation [5], [18], [19].

The concept of a head predates modern linguistic theory, but is found in current theories in the areas of syntax (when relating to phrases), morphology (when relating to word structure, especially compounding) as well as semantics (when governing meaning relations). Heads are the elements of larger constructs and dominate those constructs in structural and/or semantic respects. For example, in the phrase *high blood pressure*, the noun *pressure* is usually considered to be the head. From a semantic standpoint, it governs the semantic relations of the combination [20] such that the whole phrase is a kind of *pressure*, with *blood* and *high* being semantic dependents, the first forming a compound with *pressure* and the second being an adjectival dependent of that compound once formed [NP [JJ high] [NP [NN blood] [NN pressure]]] (note that we are using the Penn Treebank tag set [21]).

Although typically structural and semantic heads coincide, this is not always a straightforward case. Some expressions may be fully or partially idiomatic where the semantics may not follow regularities regarding headedness. Ambiguity may also arise from competing structural analyses. For example, although English compounds are normally right–headed, a *secretary general* is not a kind of *general*. Similarly, a *sexually transmitted disease clinic* is not a *disease clinic* that can be *transmitted sexually* [22]. The latter is an example of the bracketing paradox where a phrase may have multiple structural analyses (e.g. [NP [NP [ADJP *sexually transmitted*] *disease*] *clinic*] vs. [NP [ADJP *sexually transmitted*] [NP *disease clinic*]]) and either idiomatic, established meanings or contextual clues are needed to choose the correct structure [23]. Generally, compounding has been observed to be more idiosyncratic with regard to semantic heads and competing structures than syntactic combinations. This is especially the case with noun–noun compounds in English, as seen above, and is complicated further if more than two nouns are involved as is the case more frequently in technical language [24].

In phrase structure grammars [25]–[27], heads are combined with other elements to form larger phrases in a hierarchical fashion. We make use of the concept of headedness in the analysis of the internal structure of multi–word terms in order to detect their lexico–syntactic similarity that can help organize them into a hierarchy. Consider, for example, two MWTs, *acute exacerbation of chronic obstructive pulmonary disease* and *chronic obstructive pulmonary disease exacerbation*, whose parse trees are shown in Fig. 1 and 2 respectively. The fact that the noun *exacerbation* is the overall head of both phrases allows us to align the two phrases by matching their heads as well as the subphrase *chronic obstructive pulmonary disease*, which in turn allows us to infer that *acute exacerbation of chronic obstructive pulmonary disease* is a hyponym of *chronic obstructive pulmonary disease exacerbation*.

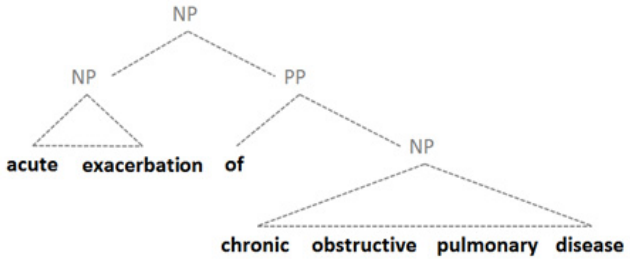

FIGURE 1. A parse tree example.

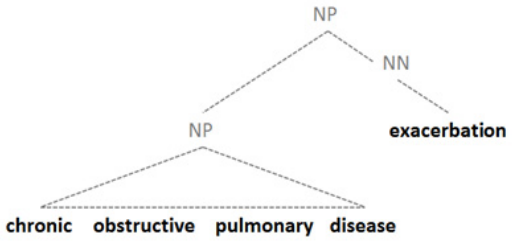

FIGURE 2. A parse tree example.

Headedness and phrase structure hierarchies have been used to extract semantic structures before; the notorious case of noun–noun compound semantics has been the subject of many recent efforts in NLP, including a special issue of Natural Language Engineering dedicated to the topic (Vol 19:3, 2013). Although the semantics of compounds remains challenging, our present focus and approach is geared toward syntactic mechanisms. In this paper, we explore the phrase structure hierarchy in an approach to measuring semantic similarity between MWTs.

We have previously developed an ATR system called FlexiTerm [8]. Given a domain–specific corpus of text documents as an input, the system outputs a list of MWTs recognized automatically. The lack of structure reduces the utility of the ATR results and limits potential applications. Hierarchy is the main organizational principle of terminologies and this is the step that we would like to automate. Hierarchical clustering is an unsupervised data mining approach that builds a hierarchy from an otherwise unstructured data set, which

makes it fit for the given purpose. The choice of a similarity measure will affect the type of hierarchy produced. Ideally, we would like it to correspond closely to the structure of taxonomy. In other words, terms representing concepts of the same type, i.e. hyponyms, co–hyponyms and hypernyms, should be grouped together in the hierarchy. We have already established that the head noun is pivotal in determining the hyponymy relation and, therefore, the similarity measure should reflect this property. In this paper, we describe our approach to extending the functionality of FlexiTerm to include measuring of semantic similarity between MWTs and, subsequently, their hierarchical clustering.

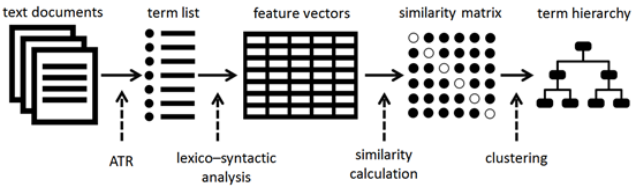

FIGURE 3. Data processing flowchart.

II. METHODS

A. METHOD OVERVIEW

Fig. 3 describes the key steps taken to organize a list of MWTs extracted automatically from a domain–specific corpus of text documents into a hierarchy based on their semantic similarity. Given a vector space, calculation of similarity and hierarchical clustering make straightforward use of existing computational methods. For them to achieve the desired outcome, the most crucial step is the choice of an appropriate vector space representation [28]. In our case, we want to marry the hierarchical nature of the internal structure of MWTs with their flat vector representation.

In our approach, this is achieved by weighing lexical features in accordance with their syntactic relation to the head noun. These relations can be extracted automatically by traditional means of syntactic analysis (dependency and constituency parsing) and modelled as a graph. The following sections provide more details on each processing step with most space dedicated to feature representation and extraction.

B. NOUN PHRASE STRUCTURE AS A DIRECTED ACYCLIC GRAPH

From the syntactic point of view, terms are noun phrases (NPs) [1], [2]. To formally represent the structure of NPs, a few definitions from graph theory are in order [29]. A *directed graph* is an ordered pair (V, A), where V is a set of elements called *vertices* and A is a subset of ordered pairs of distinct vertices called *arcs*. An arc $(u, v) \in A$ is said to *leave* its tail–vertex u and to *enter* its head–vertex v. We say that u *dominates* v, which can be denoted by $u \rightarrow v$. An *out–degree* of a vertex $u \in V$ is $d^+(u) = |\{v : (u, v) \in A\}|$, i.e. the number of arcs leaving u. A *directed acyclic graph* (DAG) is a finite directed graph that has no cycles, i.e. there is no vertex v such that there is a sequence of arcs $(v_i, v_{i+1}) \in A$ $(i = 1, \ldots, n)$ where $v_1 = v$ and $v_n = v$.

Syntactic structure of NPs can be modelled by a dependency grammar, a syntactic framework based on binary asymmetric relations, called dependencies, between individual words [30]. Dependencies reflect grammatical functions, where a word depends on another if it acts as a complement or a modifier of the latter, which in such dependency acts as the functional head. Well–formedness of a dependency structure is prescribed by four axioms [31]:

A1. One and only one element is independent.

A2. All other elements depend directly on some element.

A3. No element depends directly on more than one other.

A4. If A depends directly on B and some element C intervenes between them (in the linear order of the string), then C depends directly on A or B or some other intervening element.

Axioms A1–A3 imply that a well–formed dependency structure must be a tree, where the only independent element (i.e. the head) is its root. Axiom A4 does not allow arcs to cross in a dependency tree.

Stanford CoreNLP [32] is an NLP toolkit with a broad range of grammatical analysis tools including a dependency parser. Stanford dependencies are triplets that include the name of the relation, governor and dependent [33]. Stanford CoreNLP supports a collapsed representation of dependencies, in which dependencies involving prepositions, conjuncts and relative clauses are collapsed to get direct dependencies between content words [34]. Let us consider, for example, dependency relations within the phrase *acute exacerbation of chronic obstructive pulmonary disease*. The noun *exacerbation* is the head of the corresponding dependency tree, which has got two modifiers, the adjective *acute* and the noun *disease*, which is further modified by three adjectives *chronic*, *obstructive* and *pulmonary*. Note that there is a direct dependency between the noun complement (*exacerbation*) of a preposition (*of*) and what it modifies (*disease*). Stanford dependency parser provides an option for the collapsed dependencies to preserve a tree structure. In turn, the collapsed dependencies should represent a well–formed dependency structure as prescribed by axioms A1–A4. Fig. 4 provides a tree view of the collapsed dependencies generated by the Stanford dependency parser.

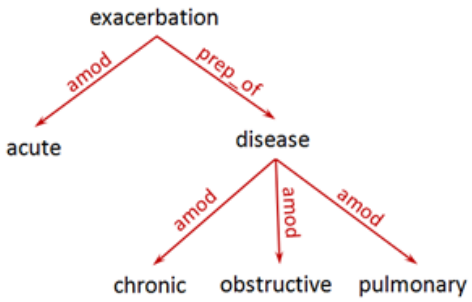

FIGURE 4. A collapsed dependency tree example.

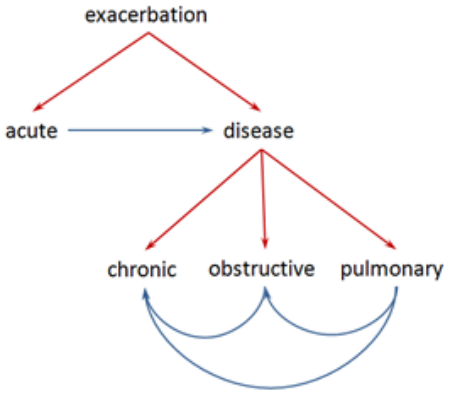

FIGURE 5. Ordering of vertices in a dependency graph for the phrase *acute exacerbation of chronic obstructive pulmonary disease.*

Every tree is a DAG. In a collapsed dependency tree, vertices are the content words and arcs are dependency relations between them. For those vertices u in the dependency tree that have out–degree $d^+(u) > 1$, we would like to add arcs between vertices linked to u while preserving a DAG structure. For example, in the tree shown in Fig. 4 $d^+(exacerbation) = 2$ and $d^+(disease) = 3$. We would like to enhance the original dependency graph by organizing the corresponding sets of vertices {*acute, disease*} and {*chronic, obstructive, pulmonary*} respectively. Specifically, we would like to induce linear order relations on these sets, e.g. *acute* $<$ *disease* and *pulmonary* $<$ *obstructive* $<$ *chronic*. Fig. 5 shows an example of a dependency graph enriched with arcs that correspond to the two linear order relations. Note that linear ordering of vertices at the same level of the tree prevents cycles from forming, thus the resulting structure is that of a DAG. Next we explain how to induce linear order in a systematic way. For this purpose, we make use of a constituency parse.

As an alternative – or rather a complement – to dependency grammar, syntactic structure of NPs can be modelled by a phrase structure grammar, a syntactic framework based on constituency relations where individual words are grouped into phrases in a hierarchical fashion [35]. Fig. 1 shows an example of a constituency parse. The tree structure of a constituency parse can be used to compare the strength of association between words.

For example, focusing on the set of descendants of the node *exacerbation* in the collapsed dependency tree shown in Fig. 4, we can order the set {*acute, disease*} using the strength of association with their parent (*exacerbation*) in the constituency parse tree shown in Fig. 1. Using the depth of the most specific common antecedent to measure the strength of association we get $S(acute, exacerbation) = 2$ and $S(disease, exacerbation) = 1$. Given that $S(acute, exacerbation) > S(disease, exacerbation)$, we conclude that the word *acute* is more strongly associated with the word *exacerbation* than is the word *disease*, and, therefore, by convention the word *acute* should come before the word *disease* in the linear order. In case of a tie, we introduce another convention based on the original word order in the given phrase.

For example, focusing on the set of descendants of the node *disease* in the collapsed dependency tree shown in Fig. 4, we can attempt to order the set {*chronic, obstructive, pulmonary*} using the strength of association with the word *disease*. Using the depth of the most specific common antecedent in the constituency parse tree to measure the strength of association we get a tie: $S(chronic, disease) = S(obstructive, disease) = S(pulmonary, disease) = 3$. Using the original word order from right to left to break the tie, we conclude that *pulmonary* $<$ *obstructive* $<$ *chronic*. If we now add the newly introduced linear order relationships to the original dependency graph given in Fig. 4, we get a DAG shown in Fig. 5.

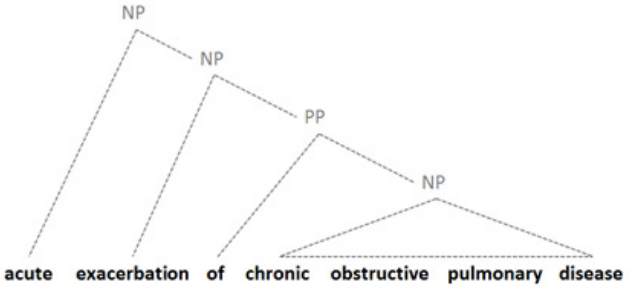

FIGURE 6. An alternative constituency parse.

If we consider an alternative constituency parse for the given phrase (see Fig. 6), we get $S(acute, exacerbation) = 1$ and $S(disease, exacerbation) = 2$. Given that $S(acute, exacerbation) < S(disease, exacerbation)$, we conclude that the word *disease* is more strongly associated with the word *exacerbation* than is the word *acute*, and, therefore, by convention the word *disease* should come before the word *acute* in the linear order. As before, the words *chronic, obstructive, pulmonary* are tied, i.e. $S(chronic, disease) = S(obstructive, disease) = S(pulmonary, disease) = 4$. Using the original word order from right to left to break the tie, we conclude that *pulmonary* $<$ *obstructive* $<$ *chronic*. If we now add the newly introduced linear order relations to the original dependency graph given in Fig. 4, we get a DAG shown in Fig. 7, which differs from the one shown in Fig. 5 only by the direction of the arc between the words *acute* and *disease*.

C. TOPOLOGICAL ORDERING OF A DIRECTED ACYCLIC GRAPH

Going back to the graph theory, a *topological ordering* of a directed graph is defined as a linear ordering of its vertices such that for every arc (u, v) in the given graph, vertex u comes before vertex v in the given ordering [29]. Using the most recent example of a directed graph shown in Fig. 7, the sequence *exacerbation, disease, pulmonary, obstructive, chronic, acute* is a topological ordering as is the sequence *exacerbation, disease, acute, pulmonary, obstructive, chronic*. Obviously, the given examples show that a

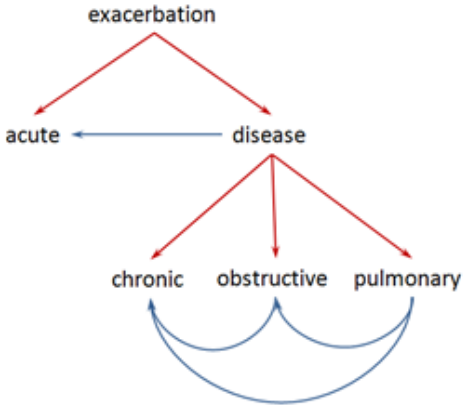

FIGURE 7. Alternative ordering of vertices in a dependency graph for the phrase *acute exacerbation of chronic obstructive pulmonary disease.*

topological ordering need not be unique. A topological ordering of a directed graph exists if and only if the graph is acyclic, in which case a topological ordering can be found in linear time [36].

In this section, we explained the manner in which dependencies between the words in an NP can be represented by a DAG. Therefore, we can find a topological ordering of the words in an NP, where the first element in the ordering is always the head noun. Adding a constraint that elements of simple NPs – the ones that contain no nested phrases (e.g. NP*chronic obstructive pulmonary disease* in the constituency parse shown in Fig. 7) – must stay adjacent in the topological ordering, we can further reduce the search space. Under these constraints, the only acceptable topological ordering of a DAG shown in Fig. 7 is the sequence *exacerbation, disease, pulmonary, obstructive, chronic, acute*. Intuitively, such ordering reflects the strength of association with the head noun.

D. THEORY VERSUS PRACTICE

There are practical challenges associated with implementing the proposed theoretical approach, which are related to the performance of constituency and dependency parsers in terms of efficiency and accuracy [23], [37]. The most prominent issue associated with parsing MWTs is that of identifying post–modifiers in NPs. The basic canonical structure of an English NP consists of a determiner (e.g. an article), a modifier (e.g. an adjective), followed by the obligatory head noun, which could be followed by a post–modifier (typically phrasal), all modifiers being entirely optional [38], [39]. In the absence of a post–modifier, the head noun will be the right–most noun in the NP. Based on this assumption, a post–modifier often gets erroneously identified as the head. For example, let us observe the differences in dependency graphs (see Fig. 8–11) obtained automatically by Stanford CoreNLP (shown on the left) against those defined manually by a linguist (shown on the right). The only correctly parsed phrase is the one illustrated in Fig. 11. In all others, the post–modifier was treated either as the head of a sub–phrase (Fig. 8) or the

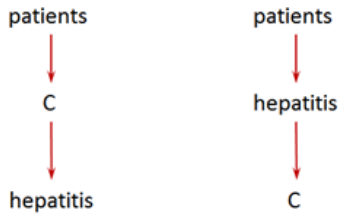

FIGURE 8. Dependency graphs for the phrase *patients with hepatitis C.*

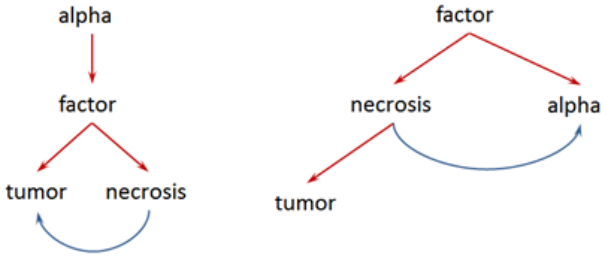

FIGURE 9. Dependency graphs for the phrase *tumor necrosis factor alpha.*

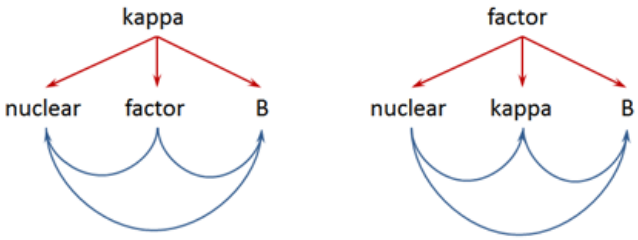

FIGURE 10. Dependency graphs for the phrase *nuclear factor kappa B.*

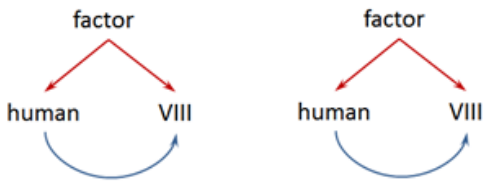

FIGURE 11. Dependency graphs for the phrase *human factor VIII.*

overall head (Fig. 9 and 10), changing the topological order of the given phrases from *patients, hepatitis, C* to *patients, C, hepatitis* in Fig. 8, from *factor, necrosis, tumor, alpha* to *alpha, factor, necrosis*, *tumor* in Fig. 9 and from *factor, nuclear, kappa, B* to *kappa, factor, nuclear*, *B* in Fig. 10.

Biomedical domain exhibits prevalent use of post–modifiers in the formation of MWTs [18], in particular in the form of numerals both Arabic (e.g. *diabetes mellitus type 2*) and Roman (e.g. *blood coagulation factor IX*), letters both Latin (e.g. *acute hepatitis B*) and Greek (e.g. *thyroid hormone receptor beta*), Latin phrases (e.g. *papillary carcinoma in situ*) or a combination of these (e.g. *human factor VIIIa* or *vitamin B12*). These modifiers are typically introduced to enumerate different varieties of the same concept so as to lexically distinguish between these instances in a

discourse. As such, these modifiers on their own usually encode little or no domain–specific meaning. For instance, the letter *B* in *nuclear factor kappa B* bears no relationship whatsoever to the same letter in *acute hepatitis B*. Incorrectly treating it as the head of the two respective phrases would give it unduly importance and could skew the lexico–syntactic comparison of the two otherwise unrelated terms. Still, this special class of modifiers cannot be treated as stop words and simply removed from consideration, because they do encode useful information when collocated with other lexical units within a MWT. Their heavy dependence on other lexical units should be reflected by the relatively low priority given to them in any lexico–syntactic comparison of the respective terms.

In summary, by parsing an NP, it can be represented as a DAG, which can be ordered in linear time using an existing algorithm. However, with the accuracy of parsers in the biomedical domain being in the low 90s at best [40], we are likely to see parsing errors translated into inaccurate topological ordering. The fact that our input is restricted to MWTs recognized automatically by FlexiTerm reduces the complexity of the parsing problem, which allowed us to implement an efficient heuristic approach to approximating topological ordering. The approach described thus far is certainly more general and remains a viable option pending future improvements in parsing performance. In the context of this study, it provides a formal mathematical description of the problem at hand.

E. ASSUMPTIONS

FlexiTerm recognizes MWTs whose structure conforms to a set of pre–defined lexico–syntactic patterns [8]. We will limit our discussion to the default set of patterns, which include:

T1. $(JJ \mid NN)^{+}\ NN$, e.g. *congestive heart failure*

T2. $(NN \mid JJ)^{*}\ NN\ POS\ (NN \mid\ JJ)^{*}\ NN$, e.g. *Hoffa's fat pad*

T3. $(NN \mid\ JJ)^{*}\ NN\ IN\ (NN \mid\ JJ)^{*}\ NN$, e.g. *acute exacerbation of chronic bronchitis*

These constraints reduce the complexity of the parsing problem. To further simplify the problem, we assume that the syntactic structure of MWTs of these three types complies with the structure shown in Fig. 12–15. Note that the provided structures assume the absence of post–modifiers. We will explain later how post–modifiers will be dealt with.

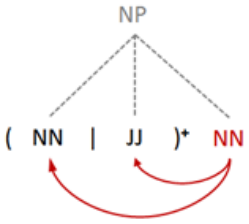

FIGURE 12. Assumed dependency and constituency parses of NPs of type T1.

For the simple NPs of types T1 and T2, minor deviations from the correct syntactic structure are irrelevant as the corresponding phrases would always be ordered from right

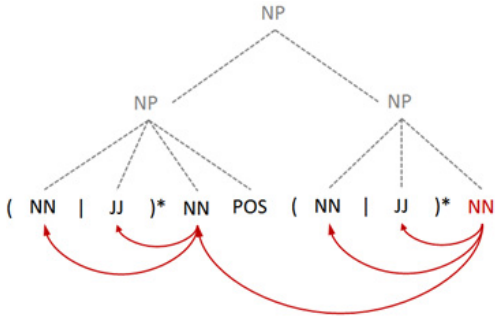

FIGURE 13. Assumed dependency and constituency parses of NPs of type T2.

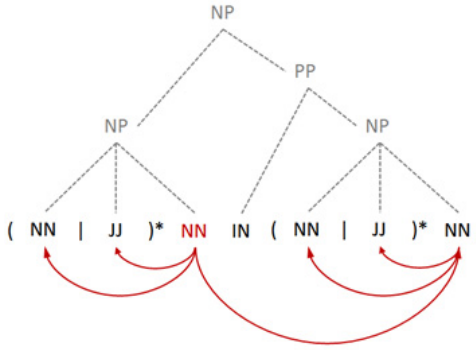

FIGURE 14. Assumed dependency and constituency parses of NPs of type T3 with any preposition other than *of*.

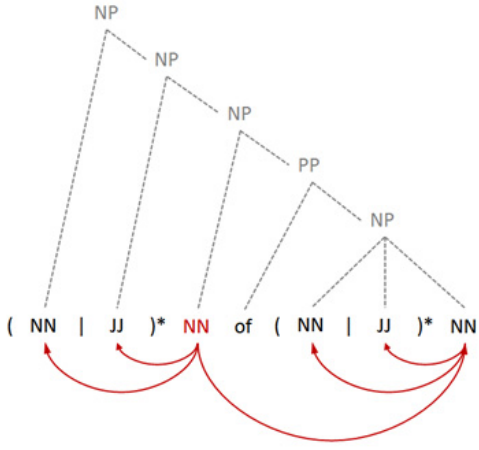

FIGURE 15. Assumed dependency and constituency parses of NPs of type T3 with the preposition *of*.

to left. For example, let us compare the correct constituency parse of the phrase *Alzheimer's disease assessment scale* [NP [NP *Alzheimer's disease*] [NP *assessment scale*]] against the assumed one [NP [NP *Alzheimer's*] [NP *disease assessment scale*]]. After removing the possessive, in both cases we get the same order – *scale, assessment, disease, Alzheimer*.

Further, we differentiate between two subtypes of complex NPs of type T3 depending on a specific preposition used. The special treatment of the preposition *of* is based on the

observation of synonyms recognized by FlexiTerm. Consider, for example, two synonyms *complete cartilage loss* and *complete loss of cartilage*(for more examples of semantic interpretation of NPs using paraphrases see [41]). Ideally, the given synonyms should have the same topological order. The simple NP variant *complete cartilage loss* has got the following order – *loss, cartilage, complete* (note that prepositions, as stop words, are not included in the order). If we assume the alternative NP variant has got the structure shown in Fig. 14, then its topological order would be *loss, complete, cartilage*. If, however, we assume it has got the structure shown in Fig. 15, then its topological order would be *loss, cartilage, complete*. In theory, this can be explained by the genitive use of the preposition *of*, where the forms NN_1 POS NN_2 and NN_2 of NN_1 are equivalent with the possessive often being omitted, albeit incorrectly. Other prepositions do not usually exhibit such strong association to the head noun, e.g. *oxygen saturation on room air* or *common migraine without aura*, where adjectival and nominal modifiers take precedence to the prepositional modifier as reflected by the structure shown in Fig. 14. There are, of course, exceptions to these rules, e.g. *range of motion exercises* whose correct parse [NP [NP *range* [PP *of motion*]] *exercises*] does not correspond to either of the proposed structures or the corresponding topological orders. Such exceptions will naturally introduce some degree of noise into the processed data. Its effects will be explored later by evaluating the end goal of this study, which is to cluster semantically similar terms.

F. A HEURISTIC APPROACH TO TOPOLOGICAL ORDERING

Previously described lexico–syntactic constraints on term formation patterns (T1–T3) together with assumptions on their syntactic structure (Fig. 12–15) allowed us to implement an efficient heuristic approach to approximating topological ordering of content words within MWTs. To effectively deal with a previously discussed class of post–modifiers, we add a constraint that no such modifier should come before a regular content word in the topological ordering. The following pseudocode provides a summary of the proposed heuristic approach:

1. Tokenize a term and add a special left–boundary token (LBT) at the start.
2. Remove the following tokens:
   a. possessives (e.g. *Hoffa'sfat pad*)
   b. past participles that follow a hyphen (e.g. *immunoreceptor tyrosine–based activation motif*)
   c. preposition *like* (e.g. *killer–cell immunoglobulin–like receptor*)
   d. punctuation (e.g. *Epstein–Barr virus*)
   e. numerals (e.g. *casein kinase II* or *24,25–dihydroxyvitamin D3*)
3. Move all tokens of the following types ahead of the left–boundary token:
   a. letters (e.g. *nuclear factor kappa B*)
   b. Latin phrases (e.g. *papillary carcinoma in situ*)
4. Remove prepositions in Latin phrases (e.g. *papillary carcinoma in situ*).
5. If a preposition is present, then let us refer to the sequence of tokens from the preposition to the right as PP.
   a. If the preposition is *of*, then move PP in front of the token that immediately precedes the preposition *of*.
   b. If any other preposition, then move PP immediately after the left boundary token.
6. Invert the order of all tokens.
7. Remove the left boundary token and any stop words.

TABLE 1. A run–through example of topological ordering

Step	Tokens	Comment
1	LBT tumor necrosis factor–alpha release in human monocytes	LBT inserted.
2	LBT tumor necrosis factor alpha release in human monocytes	Hyphen deleted.
3	alpha LBT tumor necrosis factor release in human monocytes	Greek letter moved.
4	alpha LBT tumor necrosis factor release in human monocytes	No action.
5	alpha LBT in human monocytes tumor necrosis factor release	Prepositional phrase moved.
6	release factor necrosis tumor monocytes human in LBT alpha	Order inverted.
7	release factor necrosis tumor monocytes human alpha	LBT and stop words removed.

Term: *tumor necrosis factor–alpha release in human monocytes*.

TABLE 2. A run–through example of topological ordering.

Step	Tokens	Comment
1	LBT adeno–associated in vivo gene therapy	LBT inserted.
2	LBT adeno in vivo gene therapy	Hyphen and past participle deleted.
3	in vivo LBT adeno gene therapy	Latin phrase moved.
4	vivo LBT adeno gene therapy	Preposition *in* removed.
5	vivo LBT adeno gene therapy	No action.
6	therapy gene adeno LBT vivo	Order inverted.
7	therapy gene adeno vivo	LBT removed.

Term: *Adeno–associated in vivo gene therapy*.

Tables 1 and 2 provide two run–through examples for the given algorithm. They also illustrate the motivation behind specific algorithm steps. For example, past participles are removed from hyphenated expressions as they are considered to be auxiliary in the sense that they are primarily supporting the correct syntax rather than carrying significant semantic load. Preposition *like* is removed for the same reason. In addition, *like* being a preposition, we want to exclude it from consideration in Step 5. For the same reason, we remove prepositions found within Latin phrases. The given

algorithm could easily be adapted to process more complex terms recursively, one prepositional phrase at the time from left to right. For example, *mutation in the inhibitor of kappa light polypeptide gene enhancer in B cells* would be ordered as *mutation, inhibitor, enhancer, gene, polypeptide, light, cells, kappa, B*.

G. VECTOR SPACE

A topological order of the content words that comprise a MWT allows us to assign different weights to different words based on their position in the given order. The idea is similar to that of the vector space model used in information retrieval, where text documents are represented by feature vectors. Each feature corresponds to a word and it is assigned a weight based on its relevance to the document, e.g. using a statistical measure such as term frequency–inverse document frequency [42]. In turn, vector representation allows documents to be easily compared against one another using the simple concepts of angles or distances borrowed from analytic geometry.

Going back to our original problem, let us explain how MWTs could be represented by feature vectors. Each feature corresponds to a content word w or, more precisely, its stem. Its relevance to a given MWT t, $R(w)$, is calculated as a non–negative non–increasing function $f(p(w))$ of its position p in the topological order of the term t. The proximity of two vectors can be calculated using measures such as Euclidian distance or cosine similarity. We opted for the latter because it represents a measurement of orientation and not magnitude [43]. As such, it is preferred in the context of our particular vector space representation. Namely, the proposed feature vectors will be sparse, i.e. their elements will have mostly zero values, and consequently Euclidean distance would exhibit weak discrimination in face of high dimensionality [44].

H. HIERARCHICAL CLUSTERING

Having chosen a proximity metrics in a vector space, MWTs can now be clustered using their feature vectors. In particular, hierarchical clustering can be used to organize terms into a hierarchy. In agglomerative hierarchical clustering this is achieved by iteratively merging clusters. The key operation here is the computation of the proximity between clusters. Different criteria can be used to compare two clusters, e.g. single, complete or average linkage [45], [46]. Single linkage (or minimum distance) is based on the distance between two closest members of the respective clusters. Single linkage can handle non–elliptical shapes of the clusters, but it is sensitive to noise and outliers [47]. Conversely, complete linkage (or maximum distance) is based on the distance between two furthest members of the respective clusters. Complete linkage is less susceptible to noise and outliers, but it tends to break large clusters. Finally, average linkage (or average distance) is based on the average pairwise distance between the members of the respective clusters. It represents a compromise between single and complete linkage. It is less susceptible to noise and outliers, but it is biased towards spherical clusters. Our implementation of hierarchical clustering supports all three modes of agglomeration.

The results of hierarchical clustering are often visualized using a dendrogram, a tree diagram that illustrates how clusters are iteratively merged. Leaf nodes correspond to individual elements being. Each internal node corresponds to a cluster obtained by merging the children nodes. Its height corresponds to the proximity of the merged clusters. We chose to formally encode dendrograms using the Newick format, a simple grammar that allows tree structure to be represented using parentheses and commas [48]. It also allows for storing node labels and branch lengths. The format is widely used in bioinformatics applications to store, exchange and display phylogenetic trees [49]. All dendrograms in this article have been visualized using an online tool called EvolView [50], [51].

III. RESULTS

We will describe the details of our experiments in the context of the data processing flow shown in Fig. 3. The section on raw data describes the properties of text documents used as input to ATR. The section on processed data describes the parameters of ATR and the selection of MWTs for further processing. The data representation section describes how MWTs were converted into feature vectors. In this study we make use of existing clustering methods. For them to perform well, the most crucial step is the choice of an appropriate vector space representation, which is where the main contribution of this study lies. Therefore, to evaluate the clustering performance, this is where we introduce the baseline as an alternative data representation method. Finally, the results were evaluated in terms of clustering tendency and clustering accuracy and reported in the corresponding sections.

A. RAW DATA

A study of subdomain variation in biomedical language has highlighted significant implications for evaluation of NLP tools [56]. In particular, the study emphasized that molecular biology is not representative of the overall biomedical domain, meaning that the results obtained using a corpus from this subdomain (e.g. [52]) cannot be generalized. Similarly, a comparative evaluation of term recognition approaches revealed that the choice of corpora have a significant impact on their performance [57]. Therefore, in order to evaluate our method across a wide variety of sublanguages, i.e. languages confined to specialized domains [58], we used 9 data sets associated with a range of biomedical topics and discourse types (see Table 3 for basic description).

B. PROCESSED DATA

Each data set described in Table 3 was processed by FlexiTerm [8] in order to automatically extract MWTs. The latest version of FlexiTerm integrates recognition of acronym and their mapping to the corresponding full forms into the term recognition process [59]. It supports two modes of

TABLE 3. Data sets used in evaluation.

Data set	Topic	Document type	Document number	Source
D1	molecular biology (GENIA corpus) [52]	abstract	100	PubMed
D2	chronic obstructive pulmonary disease	abstract	300	PubMed
D3	NCBI disease corpus [53]	abstract	300	PubMed
D4	study subjects [54]	methods section	400	PMC
D5	clinical trials (radiotherapy)	registration summary	100	NIH
D6	obesity [55]	discharge summary	200	i2b2
D7	knee pain	referral letter	400	NHS
D8	knee MRI scan	imaging report	500	NHS
D9	anterior cruciate ligament	user post	500	open Web

acronym recognition: (1) explicit (or local) acronyms, which are defined in a text document following scientific writing conventions, and (2) implicit (or global) acronyms, which are used in a text document without an explicit definition. Appropriate options were used for each data set, i.e. option (1) was used with scientific reports (i.e. data sets D1–D5), whereas option (2) was used with clinical narratives (i.e. data sets D6–D9). No other changes to the default values of FlexiTerm parameters were made.

FlexiTerm groups all variants of the same term together by neutralizing main sources of variation in biomedical terms – orthographic, morphological and syntactic variation as well as acronyms. In order to measure similarity between MWTs, the most frequent term variant other than the acronym was selected as the term representative. For each data set, we selected 120 top–ranked MWTs (i.e. their representatives) to conduct clustering experiments.

C. DATA REPRESENTATION

MWTs were converted into feature vectors as described previously in Section II.G. In short, content words were used as features. Given a MWT, each content word was weighed depending on its position in a topological order of all content words that comprise the MWT. In our experiments, the weight was chosen to be inversely proportional to the position. We used constant weights in an alternative data representation. Note that this is equivalent to the conventional BOW representation discussed previously in Section I. This representation was used to provide the baseline in our experiments.

D. CLUSTERING TENDENCY

Before performing clustering experiments, we assessed the clustering tendency of two data representation methods. Clustering tendency measures the degree to which a given data set exhibits a clustering structure. For example, data that contain compact non–overlapping clusters are regarded to have higher clustering tendency. On the other hand, randomly distributed data have little or no clustering tendency. It is important to assess clustering tendency because clustering methods will cluster data even in the absence of natural clusters, i.e. those whose members are sufficiently related to one another and sufficiently unrelated to non–members so as to facilitate comprehension of the ways in which individual elements are related.

We employed the visual assessment of tendency (VAT) method [60]. Given a dissimilarity matrix, whose cells express the similarity between the corresponding elements, the VAT algorithm re–orders the elements (i.e. the corresponding rows and columns of the matrix) so that more similar elements appear closer in the new ordering. Visualization of the re–ordered dissimilarity matrix can then be used to assess the degree of clustering tendency. If the data have stronger clustering tendency, then the matrix will appear to have a more prominent block–diagonal structure. In practice, each block in the matrix corresponds to a cluster present in the data. On the other hand, if the data have poorer clustering tendency, then the matrix will appear to have a less prominent block–diagonal structure.

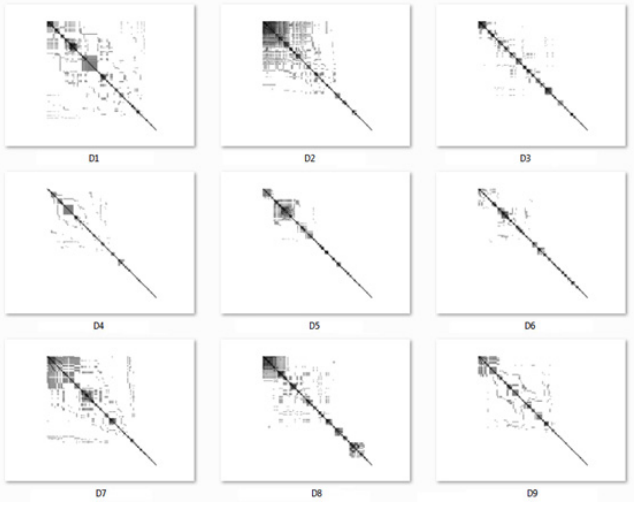

FIGURE 16. Visual assessment of clustering tendency for the baseline data representation.

We ran the VAT algorithm on both data representations. The results are shown in Fig. 16 and 17. Visual inspection of the results reveals that the proposed weighted data representation has got stronger clustering tendency than the baseline representation as illustrated by higher concentration of blocks along the diagonal and reduced randomness away from the diagonal. The next step is to check whether this change in the topology of the feature space more accurately reflects the underlying semantics. This hypothesis is tested by assessing the clustering accuracy.

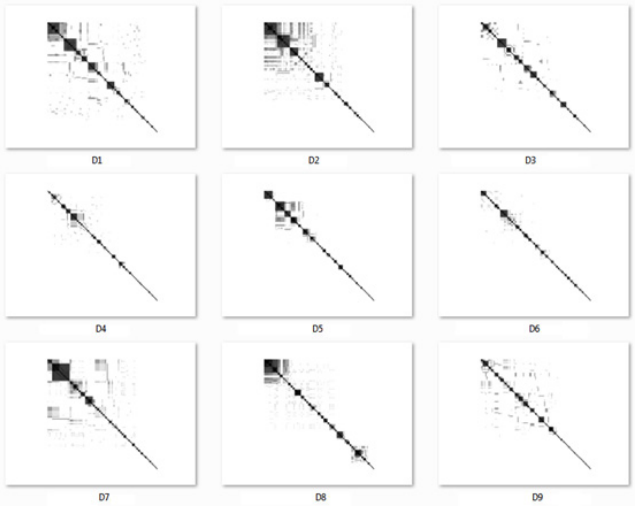

FIGURE 17. Visual assessment of clustering tendency for the weighted data representation.

E. CLUSTERING ACCURACY

Evaluating the results of hierarchical clustering is an open research problem. The approaches used in evaluating the results of partitional clustering based on measuring intra– and inter–cluster distances do not translate easily into hierarchical clustering because of the different nature of the clusters produced – the ones in partitional clustering do not overlap, whereas the ones in hierarchical clustering do. Therefore, different measures were proposed to evaluate the results of hierarchical clustering. For example, [61] proposed cutting dendrograms at various levels and counting the number of matching elements in the remaining clusters. In text mining, hierarchical clustering is often evaluated in the context of a specific application, e.g. browsing a large document collection [62]–[64].

We too considered a practical application of clustering results. Our ultimate aim was for the automatically induced hierarchy of terms to mimic the structure of a taxonomy. In other words, terms representing concepts of the same type, i.e. hyponyms, co–hyponyms and hypernyms, should be grouped together in the hierarchy. To create the gold standard, we organized 120 terms extracted from each data set into a hierarchy using the following principles: (1) All co–hyponyms should be grouped at the same level, e.g. *core binding factor alpha* and *core binding factor beta*. (2) Hyponyms should be at lower level than their hypernyms, e.g. *core binding factor*should be a level above *core binding factor alpha* and *core binding factor beta*. (3) If it does not affect conditions (1) and (2), then a term (or a cluster) should be grouped with the most related cluster of terms, e.g. *cell line* should be grouped with a cluster containing specific cells such as *B cell* or *Jurkat cell*.

To evaluate an automatically generated dendrogram, it should be compared to the gold standard. The more similar the two are, the better the clustering results. To estimate the similarity between the two hierarchies, we used the Robinson–Foulds metric [65]. Given a pair of distinct unrooted trees, each having the same set of labelled leaves, the Robinson–Foulds distance between the two trees is defined as the smallest number of contractions required to convert one tree into the other. A contraction is an operation performed on an edge by creating a union of the corresponding vertices.

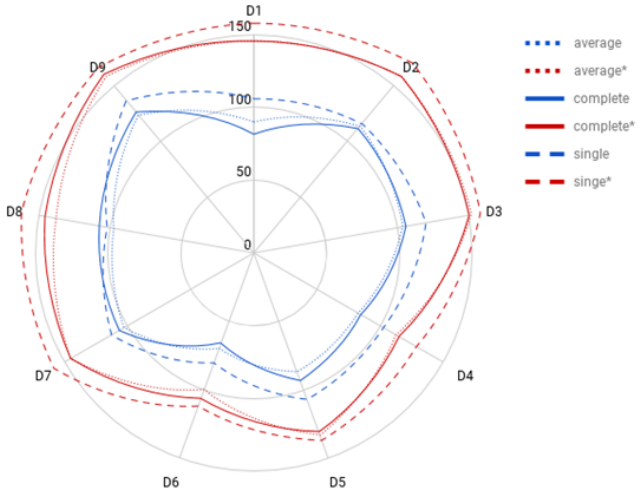

FIGURE 18. The Robson–Foulds distance from the gold standard.

Dendrograms can be viewed as a particular type of unweighted phylogenetic trees for which the Robinson–Foulds distance can be computed efficiently, i.e. in time linear to the number of leaves [66]. We used DendroPy [67] to compute the Robinson–Foulds distance with respect to the gold standard. Fig. 18 provides the values calculated for this distance for each dataset and the parameters of our experiments, which include the choice of data representation (including the baseline – indicated by an asterisk) and agglomerative method (single, complete and average linkage). Our data representation method (blue lines) outperformed the baseline (red lines) in terms of clustering accuracy regardless of the agglomerative method used. Given a data representation, there is little difference between complete and average linkage. However, with a single exception (see D8), the performance of single linkage is consistently poorer than that of the other two methods.

To illustrate the differences in clustering accuracy, we selected a few main categories of terms (e.g. in the data set D1 these would be *cells*, *proteins*, etc.) and color–coded them in the dendrograms provided in the supplementary files as well as Fig. 19 and 20 (e.g. in data set D1 *proteins* are highlighted in yellow). Visual inspection of the dendrograms demonstrates that the weighted data representation provides better consistency in grouping the terms of the same category together. This is more consistent with a taxonomic organization principle (i.e. is–a relationship), hence 30% fewer contractions are required to map dendrograms from Fig. 20 to the corresponding taxonomies than the baseline ones.

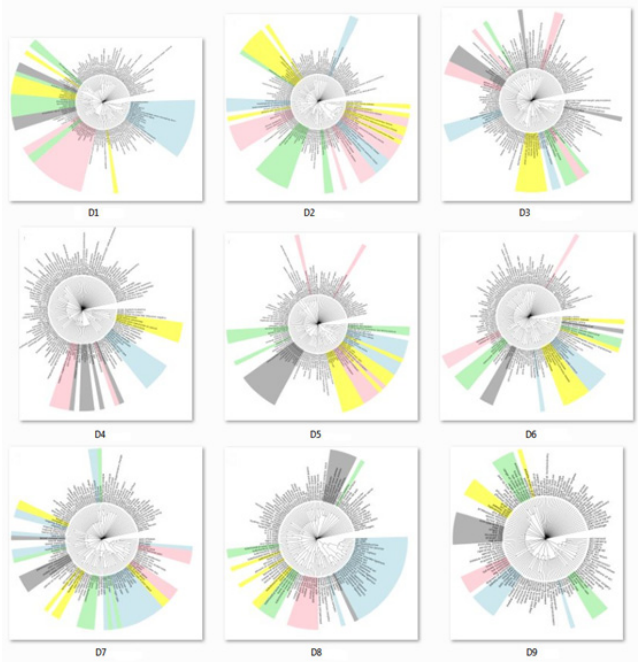

FIGURE 19. Dendrograms obtained from baseline data representation using complete linkage.

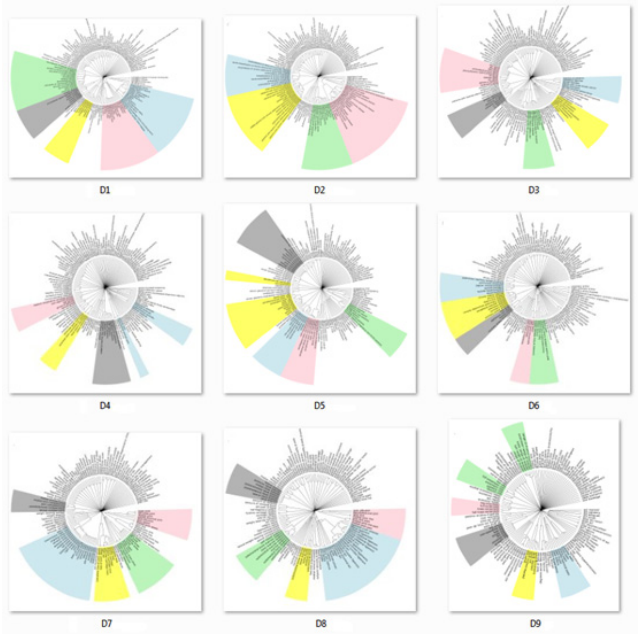

FIGURE 20. Dendrograms obtained from the weighted data representation using complete linkage.

Manual inspection of the dendrograms confirmed that, as intended, clustering on the weighted data representation tends to favor semantic similarity (based on is–a relationship) over semantic relatedness (based on any relationship between terms including but not limited to is–a) unlike the conventional BOW approach. Therefore, the weighted data representation approach is better suited for the task of automatic taxonomy construction.

IV. CONCLUSION

We presented an approach to organizing a list of MWTs extracted automatically from a domain–specific corpus of text documents into a hierarchy based on their semantic similarity. Given a vector space, calculation of similarity and hierarchical clustering make straightforward use of existing computational methods. For them to achieve the desired outcome, the most crucial step is the choice of an appropriate vector space representation, which is where the main contribution of this study lies. In our approach, we translated the graph–like structure of MWTs into a flat vector representation. To define the problem, we first formalized interpretation of the noun phrase structure based on graph theory and used it to define topological ordering of its constituents based on constituency and dependency relations between them. Given a DAG, such ordering can be found in linear time using an existing algorithm. However, this approach is sensitive to errors associated with automatically inferred constituency and dependency relations. Therefore, we implemented an alternative algorithm, which, given a noun phrase, approximates the topological ordering of its constituents. Such ordering is then used to assign different weights to different words based on their position in the ordering. Clustering performed on such vector space representation shows considerable improvement over the conventional BOW representation, i.e. it more consistently reflects semantic similarity between the terms. Semantic similarity is based on is–a relationship, which represent the main organizational principle of terminologies. Therefore, in combination with our existing term recognition approach, the method described in this study can be used to semi–automate taxonomy construction from a corpus of domain–specific documents. Our approach is complementary to distributional semantics approaches (e.g. [68], [69]), which require large amounts of contextual information to infer relations between terms, in the sense that it uses the terms themselves to make comparisons. The fact that our approach does not require a large data set to make such inferences is advantageous in scenarios where accessibility of text data is limited, e.g. in clinical applications where privacy concerns exist.

REFERENCES

[1] B. Faille, "Study and implementation of combined techniques for automatic extraction of terminology," in *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, P. Resnik and J. Klavans, Eds. Cambridge, MA, USA: MIT Press, 1996, pp. 49–66.

[2] K. Kageura and B. Umino, "Methods of automatic term recognition: A review," *Terminology*, vol. 3, no. 2, pp. 259–289, 1996.

[3] K. Frantzi and S. Ananiadou, "Automatic term recognition using contextual cues," in *Proc. 3rd DELOS Workshop Cross-Lang. Inf. Retr.*, Zurich, Switzerland, 1997, pp. 2155–2162.

[4] C. Jacquemin, *Spotting and Discovering Terms through Natural Language Processing*. Cambridge, MA, USA: MIT Press, 2001.

[5] O. Bodenreider, A. Burgun, and T. C. Rindflesch, "Lexically-suggested hyponymic relations among medical terms and their representation in the UMLS," in *Proc. 9th Int. Conf. Terminol. Artif. Intell.*, Paris, France, 2001, pp. 11–21.

[6] I. Spasić, S. Ananiadou, J. McNaught, and A. Kumar, "Text mining and ontologies in biomedicine: Making sense of raw text," *Briefings Bioinf.*, vol. 6, no. 3, pp. 239–251, 2005.

[7] A. Cruse, *Lexical Semantics*. Cambridge, U.K.: Cambridge Univ. Press, 1986.
[8] I. Spasić, M. Greenwood, A. Preece, N. Francis, and G. Elwyn, "FlexiTerm: A flexible term recognition method," *J. Biomed. Semantics*, vol. 4, 27, 2013.
[9] Z. Kozareva and E. Hovy, "A semi-supervised method to learn and construct taxonomies using the Web," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Cambridge, MA, USA, 2010, pp. 1110–1118.
[10] P. Velardi, S. Faralli, and R. Navigli, "OntoLearn reloaded: A graph-based algorithm for taxonomy induction," *Comput. Linguistics*, vol. 39, no. 3, pp. 665–707, 2013.
[11] M. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proc. 14th Int. Conf. Comput. Linguistics (COLING)*, Nantes, France, 1992, pp. 539–545.
[12] S. A. Caraballo, "Automatic construction of a hypernym-labeled noun hierarchy from text," in *Proc. 37th Annu. Meeting Assoc. Comput. Linguistics Comput. Linguistics*, College Park, MD, USA, 1999, pp. 120–126.
[13] S. Cederberg and D. Widdows, "Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction," in *Proc. 7th Conf. Natural Lang. Learn.*, 2003, pp. 111–118.
[14] K. Shinzato and K. Torisawa, "Acquiring hyponymy relations from Web documents," in *Proc. Hum. Lang. Technol. Conf./North Amer. Chapter Assoc. Comput. Linguistics Annu. Meeting*, Boston, MA, USA, 2004, pp. 73–80.
[15] E. Morin and C. Jacquemin, "Automatic acquisition and expansion of hypernym links," *Comput. Humanities*, vol. 38, no. 4, pp. 363–396, 2004.
[16] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2004, pp. 1297–1304.
[17] Z. Kozareva, E. Riloff, and E. H. Hovy, "Semantic class learning from the Web with hyponym pattern linkage graphs," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, Columbus, OH, USA, 2008, pp. 1048–1056.
[18] A. T. McCray and A. C. Browne, "Discovering the modifiers in a terminology data set," in *Proc. Amer. Med. Inform. Assoc. Symp.*, Orlando, FL, USA, 1998, pp. 780–784.
[19] O. Bodenreider, A. Burgun, and T. Rindflesh, "Assessing the consistency of a biomedical terminology through lexical knowledge," *Int. J. Med. Inform.*, vol. 67, nos. 1–3, pp. 85–95, 2002.
[20] A. Cruse, *Meaning in Language: An Introduction to Semantics and Pragmatics*. London, U.K.: Oxford Univ. Press, 2011.
[21] M. P. Marcus and M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," *Comput. Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
[22] A. Carstairs-McCarthy, *An Introduction to English Morphology: Words and Their Structure*. Edinburgh, U.K.: Edinburgh Univ. Press, 2001.
[23] D. Vadas and J. R. Curran, "Parsing noun phrases in the Penn Treebank," *Comput. Linguistics*, vol. 37, no. 4, pp. 753–809, 2011.
[24] P. Nakov, "On the interpretation of noun compounds: Syntax, semantics, and entailment," *Natural Language Eng.*, vol. 19, no. 3, pp. 291–330, 2013.
[25] R. Borsley, *Modern Phrase Structure Grammar*. Hoboken, NJ, USA: Wiley, 1996.
[26] N. Fukui, "Phrase structure," in *The Handbook of Contemporary Syntactic Theory*, M. Baltin and C. Collins, Eds. Oxford, U.K.: Blackwell, 2003, pp. 374–406.
[27] I. A. Sag, T. Wasow, and E. M. Bender, *Syntactic Theory: A Formal Introduction*, 2nd ed. Chicago, IL, USA: Univ. of Chicago Press, 2003.
[28] C. Wang and W. Wang, "Using term clustering and supervised term affinity construction to boost text classification," in *Proc. 9th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, Hanoi, Vietnam, 2005, pp. 813–819.
[29] J. Bang-Jensen and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 2nd ed. Berlin, Germany: Springer, 2009.
[30] R. Debusmann, "An introduction to dependency grammar," Hausarbeit Hauptseminar Dependenzgrammatik SoSe, Univ. Saarlandes, Saarbrücken, Germany, Tech. Rep. 99, 2000.
[31] J. J. Robinson, "Dependency structures and transformation rules," *Language*, vol. 46, no. 2, pp. 259–285, 1970.
[32] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, Baltimore, MD, USA, 2014, pp. 55–60.
[33] M.-C. de Marneffe *et al.*, "Universal Stanford dependencies: A cross-linguistic typology," in *Proc. 9th Int. Conf. Lang. Resour. Eval. Lang. Resour. Eval.*, Reykjavik, Iceland, 2014, pp. 4585–4592.
[34] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2016.
[35] N. Chomsky, *Syntactic structures*, The Hague, The Netherlands: Mouton, 1957.
[36] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
[37] D. M. Cer, M.-C. de Marneffe, D. Jurafsky, and C. D. Manning, "Parsing to stanford dependencies: Trade-offs between speed and accuracy," in *Proc. 7th Int. Conf. Lang. Resour. Eval.*, Valletta, Malta, 2010, pp. 1628–1632.
[38] D. Biber, E. Finegan, S. Johansson, S. Conrad, and G. Leech, *Longman Grammar of Spoken and Written English*. London, U.K.: Longman, 1999.
[39] M. Liberman and R. Sproat, "The stress and structure of modified noun phrases in English," in *Lexical Matters*, I. Sag and A. Szabolcsi, Eds. Stanford, CA, USA: CSLI Publications, 1992, pp. 131–181.
[40] R. Cohen and M. Elhadad, "Syntactic dependency parsers for biomedical-NLP," in *Proc. Annu. Symp. Amer. Med. Inform. Assoc.*, Chicago, IL, USA, 2012, pp. 121–128.
[41] P. Nakov and M. A. Hearst, "Semantic interpretation of noun compounds using verbal and other paraphrases," *ACM Trans. Speech Lang. Process.*, vol. 10, no. 3, 2013, Art. no. 13.
[42] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1986.
[43] D. Jurafsky, and J. H. Martin, *Speech and Language Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.
[44] M. E. Houle, H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" in *Proc. 22nd Int. Conf. Sci. Statist. Database Manage.*, Heidelberg, Germany, 2010, pp. 482–500.
[45] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[46] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[47] L. Ertöz, M. Steinbach, and V. Kumar, "A new shared nearest neighbor clustering algorithm and its applications," in *Proc. 2nd SIAM Int. Conf. Data Mining*, Arlington, VA, USA, 2002, pp. 105–115.
[48] J. Archie *et al.* (1986). *The Newick Tree Format*. [Online]. Available: http://evolution.genetics.washington.edu/phylip/ newicktree.html
[49] L. Czech, J. Huerta-Cepas, and A. Stamatakis, "A critical review on the use of support values in tree viewers and bioinformatics toolkits," *Molecular Biol. Evol.*, vol. 34, no. 6, pp. 1535–1542, 2017.
[50] H. Zhang, S. Gao, M. J. Lercher, S. Hu, and W.-H. Chen, "EvolView, an online tool for visualizing, annotating and managing phylogenetic trees," *Nucleic Acids Res.*, vol. 40, no. W1, pp. W569–W572, 2012.
[51] Z. He, H. Zhang, S. Gao, M. J. Lercher, W.-H. Chen, and S. Hu, "EvolView v2: An online visualization and management tool for customized and annotated phylogenetic trees," *Nucleic Acids Res.*, vol. 44, no. W1, pp. W236–W241, 2016.
[52] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus—A semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, pp. i180–i182, Jul. 2003.
[53] R. I. Doğana, R. Leamana, and Z. Lua, "NCBI disease corpus: A resource for disease name recognition and concept normalization," *J. Biomed. Inform.*, vol. 47, pp. 1–10, Feb. 2014.
[54] D. Demner-Fushman and J. G. Mork, "Extracting characteristics of the study subjects from full-text articles," in *Proc. Annu. Symp. Amer. Med. Inform. Assoc.*, San Francisco, CA, USA, 2015, pp. 484–491.
[55] Ö. Uzuner, "Recognizing obesity and comorbidities in sparse data," *J. Amer. Med. Inform. Assoc.*, vol. 16, no. 4, pp. 561–570, 2009.
[56] T. Lippincott, D. Ó. Séaghdha, and A. Korhonen, "Exploring subdomain variation in biomedical language," *BMC Bioinf.*, vol. 12, p. 212, May 2011.
[57] Z. Zhang, J. Iria, C. Brewster, and F. Ciravegna, "A comparative evaluation of term recognition algorithms," in *Proc. 6th Int. Conf. Lang. Resour. Eval.*, Marrakech, Morocco, 2008, pp. 2108–2111.
[58] Z. Harris, "Discourse and sublanguage," in *Sublanguage: Studies of Language in Restricted Semantic Domains*, R. Kittredge and J. Lehrberger, Eds. Berlin, Germany: Walter de Gruyter, 1982, pp. 231–236.
[59] I. Spasić, "Acronyms as an integral part of multi-word term recognition—A token of appreciation," *IEEE Access*, vol. 6, pp. 8351–8363, 2018.
[60] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, USA, 2002, pp. 2225–2230.

[61] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Statist. Assoc.*, vol. 78, no. 383, pp. 553–569, 1983.
[62] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Diego, CA, USA, 1999, pp. 16–22.
[63] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, McLean, VA, USA, 2002, pp. 515–524.
[64] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, Arlington, VA, USA, 2006, pp. 357–366.
[65] D. F. Robinson and L. R. Foulds, "Comparison of phylogenetic trees," *Math. Biosci.*, vol. 53, nos. 1–2, pp. 131–147, 1981.
[66] W. H. E. Day, "Optimal algorithms for comparing trees with labeled leaves," *J. Classification*, vol. 2, no. 1, pp. 7–28, 1985.
[67] J. Sukumaran and M. T. Holder, "DendroPy: A Python library for phylogenetic computing," *Bioinformatics*, vol. 26, no. 12, pp. 1569–1571, 2010.
[68] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.
[69] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 1532–1543.

IRENA SPASIĆ received the Ph.D. degree in computer science from the University of Salford, U.K., in 2004. Following posts at the Universities of Belgrade, Salford and Manchester, she joined the Cardiff School of Computer Science and Informatics in 2010, and became a Full Professor in 2016. She leads the text and data mining research theme at Cardiff University, and is a co-founder of the U.K. Healthcare Text Analytics Research Network. Her research interests include text mining, knowledge representation, machine learning, and information management with applications in healthcare, life sciences, and social sciences.

PADRAIG CORCORAN is currently a Lecturer with the School of Computer Science and Informatics, Cardiff University. He received the European Marie Curie International Outgoing Fellowship. He spent the outgoing phase of the fellowship at the Massachusetts Institute of Technology. During the incoming phase, he was with University College Dublin.

ANDREI GAGARIN received the Ph.D. degree in computer science from the University of Manitoba in 2003. He was with the University of Quebec, Montreal, Acadia University, and the University of London. He has been a Lecturer in mathematics with Cardiff University since 2016. His main research interests are in graph theory, optimization in networks, combinatorics, operational research, data analysis, algorithms design and engineering, and workflows and access control.

ANDREAS BUERKI received the Ph.D. degree in general linguistics from the University of Basel in 2013. He is currently a Lecturer in linguistics with Cardiff University, specializing in phraseology and corpus linguistics and quantitative approaches to linguistic structure and language change. He is a member of the advisory council of the European Society of Phraseology.