Journal of Biomedical Informatics 43 (2010) 762–773

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Disambiguation of ambiguous biomedical terms using examples generated from the UMLS Metathesaurus

Mark Stevenson *, Yikun Guo

Natural Language Processing Group, Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom

ARTICLE INFO

Article history:
Received 25 November 2009
Available online 10 June 2010

Keywords:
Natural Language Processing
NLP
Word Sense Disambiguation
WSD
Unified Medical Language System
UMLS

ABSTRACT

Researchers have access to a vast amount of information stored in textual documents and there is a pressing need for the development of automated methods to enable and improve access to this resource. Lexical ambiguity, the phenomena in which a word or phrase has more than one possible meaning, presents a significant obstacle to automated text processing. Word Sense Disambiguation (WSD) is a technology that resolves these ambiguities automatically and is an important stage in text understanding. The most accurate approaches to WSD rely on manually labeled examples but this is usually not available and is prohibitively expensive to create. This paper offers a solution to that problem by using information in the UMLS Metathesaurus to automatically generate labeled examples. Two approaches are presented. The first is an extension of existing work (Liu et al., 2002 [1]) and the second a novel approach that exploits information in the UMLS that has not been used for this purpose. The automatically generated examples are evaluated by comparing them against the manually labeled ones in the NLM-WSD data set and are found to outperform the baseline. The examples generated using the novel approach produce an improvement in WSD performance when combined with manually labeled examples.

1. Introduction

The number of documents relevant to biomedical science and related areas is growing at an ever increasing rate, making it difficult for researchers and practitioners to keep track of recent developments [2]. Automated methods for cataloguing, searching and navigating these documents would be of great benefit and it has been shown that providing access to on-line medical information improves decisions made by medical practitioners [3] and consumers [4].

However, lexical ambiguity, the phenomenon where a term (word or phrase) has more than one potential meaning, makes the automatic processing of text difficult. For example, “cold” has several possible meanings in the Unified Medical Language System (UMLS) Metathesaurus [5] including “common cold”, “cold sensation” and “cold temperature”. Ambiguous terms are common in biomedical documents. Weeber et al. [6] analyzed Medline abstracts and found that 11.7% of phrases were ambiguous relative to the UMLS Metathesaurus. The NLM Indexing Initiative [7] attempted to index biomedical journals with concepts from the UMLS Metathesaurus and concluded that lexical ambiguity was the biggest challenge in the automation of this process. An information extraction system originally designed to process radiology reports encountered problems with ambiguity when it was applied to more general biomedical texts [8]. During the development of an automated knowledge discovery system Weeber et al. [9] found that it was necessary to resolve the ambiguity in the abbreviation MG (which can mean ‘magnesium’ or ‘milligram’) in order to replicate a well-known literature-based discovery concerning the role of magnesium deficiency in migraine headaches [10].

The process of resolving lexical ambiguities, Word Sense Disambiguation (WSD), is regarded as an important part of the process of understanding natural language texts [11–13]. It is necessary for applications such as information extraction and text mining which are important in the biomedical domain for tasks such as automated knowledge discovery. Several studies have shown that the best WSD performance is obtained from systems based on supervised learning approaches [12–14]. These approaches require labeled training data, examples of ambiguous terms annotated with the correct meaning. It has also been shown that the performance of supervised approaches tends to improve with access to more labeled training data [15,16] so it is important to ensure that enough examples can be obtained to provide the best performance. However, labeled training data are often not available and the majority of existing resources contain only limited numbers of examples and do not reflect the ambiguities that occur within biomedical sciences. The only resource specific to this domain, the NLM-WSD corpus (see Section 4.1), contains 100 examples for 50

* Corresponding author.
E-mail addresses: m.stevenson@dcs.shef.ac.uk (M. Stevenson), y.guo@sheffield.ac.uk (Y. Guo).

1532-0464/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2010.06.001

ambiguous domain-specific terms. But labeled training data are also extremely time-consuming and expensive to create [17,18] and it has been estimated that approximately 3.2 million sense tagged examples would be required to train a high-performance WSD system [16]. The manual labeling process is more difficult for specific domains, like biomedicine, since technical usages can only be identified by domain experts, making the process of recruiting annotators more difficult.

The costly process of manual labeling can be avoided using techniques that generate labeled examples automatically, a process that has been referred to as **pseudo-labeling** [19]. This paper describes two approaches for pseudo-labeling examples of ambiguous terms in the biomedical domain using various types of information from the UMLS Metathesaurus.

Previous approaches to pseudo-labeling are reviewed in Section 2. The two approaches based on the UMLS Metathesaurus that are used in this paper are described in Section 3. These approaches are used to generate pseudo-labeled examples for a set of 18 ambiguous terms. These examples are evaluated by using them as training data for a supervised WSD system (Section 4) and by combining them with manually labeled examples (Section 5).

2. Previous approaches to pseudo-labeling

Several approaches have been suggested for automatically generating sense tagged examples. One makes use of the fact that different senses of ambiguous words often have different translations [20,21]. For example, the word "drug" is translated to French as "médicament" when it is used to mean 'medicine' and "droguer" when it means 'narcotic'. If text and its associated translation (known as "parallel text") are available it can be used to generate sense tagged examples with the alternative translations acting as sense labels. However, the alternative translations do not always correspond to the sense distinctions in the original language and parallel text is normally difficult to obtain.

An alternative technique that does not require parallel text but relies on a lexical knowledge base has also been suggested [22]. This used WordNet [23], a lexicon that is widely used in Natural Language Processing research. The approach is based on the observation that some terms in a lexicon occur only once and, consequently, there is no doubt about their meaning. These terms are referred to as "monosemous". However, the majority of terms have more than one possible meaning, in other words they are polysemous, and the challenge is to identify examples of the term being used with a particular meaning that can be used as training data. They suggest finding the closest related sense that is monosemous as a substitute for the ambiguous term. Sentences containing the monosemous relative are identified and the relative substituted with the ambiguous term. This approach is referred to as **monosemous relatives**. For example, the term "church" can mean 'building' ("the *church* was empty") and 'institution' ("The Catholic *Church* is the largest religious body in the United States") [24]. Monosemous relatives of the 'building' meaning include 'church building', 'house of prayer' and 'synagogue'. Examples of these terms are collected and the monosemous term substituted with the polysemous one. For example, if the sentence "The synagogue is on the left at the first light" was retrieved it would be adapted to "The church is on the left at the first light" and used as an example of the 'building' meaning of "church".

A variant of the monosemous relatives approach has been applied to the biomedical domain [1]. The UMLS Metathesaurus, rather than WordNet, was used to generate monosemous relatives. Terms related to the ambiguous term are identified from the Metathesaurus and unambiguous strings associated with these concepts used as the monosemous relatives. The approach was evaluated using a corpus of 35 ambiguous abbreviations from biomedical documents. They reported precision of 96.8% but recall of just 50.6%. The low recall figure indicates that the approach could only be used to generate pseudo-labeled examples for around half of the ambiguous abbreviations in the study, although the high precision score also shows that the examples that could be generated were very useful for disambiguation. There is also evidence that abbreviations are simpler to disambiguate than other ambiguous terms [25].

A variation of this approach based on semantically similar terms rather than monosemous relatives was recently proposed [19]. Terms that are semantically similar to the ambiguous word are identified using an information-theoretic algorithm for computing distributional similarity [26]. The terms identified by this process are not associated with any particular sense of the ambiguous word so an unsupervised WSD algorithm [27] is used to identify the most probable one. For example, similar terms for "church" might include "cathedral", "chapel", "congregation", "parish" and "synagogue". If we assume that the WSD algorithm identifies "cathedral", "chapel" and "synagogue" as being related to the 'building' sense then examples of these terms would be identified and the relevant terms substituted with "church" to provide examples of that meaning.

A semi-supervised approach to the problem has also been applied to the biomedical domain [28]. They used techniques for Information Retrieval to analyze sense tagged examples and automatically download similar ones from Medline. They found that adding these new examples led to a small but significant improvement in the performance of their WSD system. The main problem with this approach is that it still relies on sense tagged examples. They used 100 such examples for each ambiguous term for these experiments [28].

Each of these approaches relies on external resources (parallel text, a domain ontology or an existing set of sense tagged examples). When processing biomedical text the domain ontology is the most convenient to obtain, since the UMLS is readily available. We present two pseudo-labeling approaches that used the UMLS Metathesaurus. The first of these is an extension of the "monosemous relatives" approach [1,22] and the second a novel approach that uses information about co-occurring concepts.

3. Generating sense tagged examples using the UMLS Metathesaurus

This section describes two methods for pseudo-labeling using the UMLS Metathesaurus. The first is an extension of the monosemous relatives approach (Section 3.2) and the second a novel approach that makes use of co-occurrence information (Section 3.3). Before describing the details of these approaches, the resources they make use of are described.

3.1. Resources used

3.1.1. UMLS Metathesaurus

The Unified Medical Language System (UMLS) [5] is a collection of controlled vocabularies related to biomedicine and contains a wide range of information that can be used for Natural Language Processing. The 2007AB version of the UMLS was used for the experiments described in this paper. This version was chosen since we had access to a mapping between the concepts in the UMLS and the senses in the data set we used for evaluation (see Section 4.1). Such a mapping is only required to evaluate our approach. It would be possible to use the approaches described in this paper to generate pseudo-labeled data with any version of the UMLS.

The mapping from the 2007AB version of the UMLS was created with the assistance of publicly available software and was manually verified.[1] A mapping between the 1999 version of the UMLS and our evaluation data is also available. We carried out experiments using the 1999 version. We found that there was little difference between the results when this version of the UMLS was used and focused our attention on carrying out a more thorough analysis using the more recent 2007AB version.[2]

The UMLS comprises three parts: Specialist Lexicon, Semantic Network and Metathesaurus. The work described in this paper uses the Metathesaurus. The Metathesaurus forms the backbone of the UMLS and is created by unifying over 100 controlled vocabularies and classification systems. It is organized around concepts. Each concept represents a meaning and is assigned a Concept Unique Identifier (CUI). For example, the following CUIs are all associated with the term "cold":

C0009443 'Common Cold'.
C0009264 'Cold Temperature'.
C0234192 'Cold Sensation'.

Each concept is also associated with at least one term and each term is labeled with a LUI. LUIs represent a range of lexical variants for a particular term.[3] For example, the possible LUIs for CUI C0009443 include L0009443: "common cold", L0001336: "Acute Nasopharyngitis", L0277994: "Acute rhinitis" and L0018673: "Head cold".

The Metathesaurus also contains a wide range of information about LUIs, CUIs and the relations between them in the form of database tables. We describe the ones that are used by our approach. The MRCON table lists the possible terms (LUIs) for each concept. Any LUIs which are linked to multiple CUIs in this table are considered to be ambiguous and are listed in the AMBIGLUI table. (For example L0277994: "Acute rhinitis" is listed in this table since it is associated with the CUIs C0009443: "Common cold" and C0086066: "Acute Coryza".) Co-occurrence relations between CUIs are found in the MRCOC table. This information is obtained through analysis of the various UMLS sources. They exist between similar concepts (e.g. "Atrial Fibrillation" and "Arrhythmia") or different concepts which share an important connection (e.g. "Atrial Fibrillation" and "Digoxin"). Although the MRCOC table lists a large number of co-occurrence relations most Metathesaurus concepts do not have any co-occurrence relations associated with them.

The MRREL table lists relations between CUIs found in the various sources that are used to form the Metathesaurus. This table lists a range of different types of relations and we use the majority of them.[4] We make use of CUIs that are connected directly, through a single relation in the MRREL table, and indirectly, via a path containing some number of intermediate concepts. For example, the CUI C0600072 "Feeding and dietary regimes" is related to C1442959 "Nutrition function" in the MRREL table by the RO "related, other" relation. We refer to these direct relations, which can be identified by examining a single row in the MRREL table, as "level 1" relations. We also refer to C0600072 and C1442959 as being "level 1 relatives". C060072 is also indirectly related to C0588458 "cholesterol reduction program"; C0600072 is related to C1442959 which is in turn related to C0588458 by the CHD "child" relation. (We allow indirect relations to be connected by paths containing different types of relations since we found that this could produce useful pseudo-labeled examples.) Relations like this are referred to as "level 2" relations since information from two rows of the MRREL table is required to identify them. C060072 and C0588458 are also referred to as being "level 2 relatives". It would also be possible to identify more distant relatives by creating paths between pairs of CUIs that include more than one intermediate concept, these could be identified by examining additional rows in the MRREL table. However, in practice we found that these were not useful for our approach and we did not make use of these more distant relatives.

3.1.2. Medline and Entrez

Medline is a database of publications in biomedical and life sciences that contains over 19 million citations[5] and indexes over 5000 academic journals.[6] It can be accessed over the internet via the Entrez[7] and PubMed[8] interfaces. Abstracts for the majority of Medline citations are freely available through these interfaces, although the full text of the article may not be available.

The experiments described here use the Entrez Programming Utilities. These provide the facility for searching Medline using Boolean queries with a wide range of options to control the abstracts that are returned. These searches can be executed automatically using a SOAP architecture.

3.1.3. Medical Subject Headings (MeSH)

Medical Subject Headings (MeSH) [29] is a controlled vocabulary for indexing biomedical and health-related documents. The 2009 version of MeSH contains over 25,000 terms organized into an 11 level hierarchy. MeSH terms are manually assigned to Medline abstracts by human indexers.

3.2. Approach 1: monosemous relatives

The monosemous relatives approach (Section 2) has been shown to generate pseudo-labeled examples that can be used to train a WSD system [1]. It only considers direct relatives of ambiguous concepts to generate pseudo-labeled examples but, since some concepts do not have direct relatives that are monosemous, it cannot generate examples for all ambiguous terms. Our first approach extends the existing approach by also making use of indirect relatives which has been shown to be a useful way of generating pseudo-labeled examples [24].

The monosemous relatives approach can be adapted for use with the UMLS in a straightforward way. The possible lexicalisations of CUIs can be found in the MRCON table as LUIs. The AMBIGLUI table lists LUIs that are ambiguous between Metathesaurus concepts. We consider any LUIs that are not mentioned in this file to be unambiguous, or "monosemous". For example, one of the

[1] We are grateful to Bridget McInnes for providing this software, which is available from http://wsd.nlm.nih.gov, and assisting in verification and correction of the mappings it suggested.

[2] It would also have been interesting to experiment with the most recent version of the UMLS. However, the process of creating a reliable mapping is time-consuming and we did not expect using the most recent version to produce any additional insights into our approach.

[3] The Metathesaurus includes other objects, including atoms and strings, but since they are not central to our approach we do not describe them here.

[4] These are CHD "child", PAR "parent", RB "related, broader", RN "related, narrower", RL "related, alike", RO "related, other" and QB "qualified by". Two relations, SIB "has sibling" and AQ "allowed qualifier", are not used since we found that the pseudo-labeled examples generated using them were not useful.

[5] http://www.nlm.nih.gov/bsd/revup/revup_pub.html

[6] http://www.nlm.nih.gov/bsd/num_titles.html

[7] http://www.ncbi.nlm.nih.gov/sites/gquery

[8] http://www.ncbi.nlm.nih.gov/pubmed/

meanings of the term "nutrition" is C0028707 'Science of Nutrition' which, by the `MRCON` table, is connected to the string 'nutrition science'. The `AMBIGLUI` table reveals that this string is not associated with any other CUIs. It is therefore considered to be monosemous and can be used to identify examples of the CUI.

However, there may be cases where the LUIs associated with a particular CUI are ambiguous or where examples of a particular string cannot be found. In these cases we make use of the CUI's relatives, starting with the direct (i.e. level 1) relatives. For example, C0028707 is directly related to C0012155 "Dietetics" in the `MRREL` table by the `CHD` "child" relation. C0012155 is associated with the monosemous string 'dietetics' which can be used to generate pseudo-labeled examples of C0028707. Indirect relatives can also be used. For example, another possible meaning of "nutrition", C06000072 "Feeding and dietary regimes", is indirectly related to C0588458 (see Section 3.1.1). C0588458 is associated with the string "cholesterol reduction program" which is monosemous and can be used to generate pseudo-labeled examples of C0600072. For example, one Medline abstract contains the phrase "Randomized controlled trial of a nonpharmacologic *cholesterol reduction program* at the worksite" which can be converted to "Randomized controlled trial of a nonpharmacologic *nutrition* at the worksite" to act as an example of C0600072. (Examples generated using this approach do not necessarily have exactly the same meaning as the sentence containing the monosemous relative and may also sound unnatural. However, it has been shown [24] that they can still be useful training data for WSD systems.)

We begin with an informal description of the process we use to generate pseudo-labeled examples and then provide a more formal description. Our approach begins with a CUI, *c*, and aims to generate a pre-defined number of pseudo-labeled examples of that CUI, which we refer to as *target*. (Section 4.1 describes how the value for *target* is chosen for each CUI.) Strings that are associated with *c* are identified and any monosemous ones used to form a query to retrieve abstracts from Medline. Any abstracts identified by this query are stored to be used as pseudo-labeled examples of *c*. If the required number of examples (i.e. *target*) have been identified the process stops. Otherwise it continues by making use of relatives of *c* identified from the `MRREL` table. The order in which the relatives of *c* are considered is based on their distance from *c*, starting with the level 1 relatives and incrementally increasing the level. We found that expanding the search beyond level 2 relatives led to CUIs that are not closely enough related to *c* being used as monosemous relatives. These more distantly related relatives did not generate useful pseudo-labeled examples. Consequently only level 1 and level 2 relatives are used.

This process is essentially a breadth first search through the UMLS Metathesaurus and is shown in Algorithm 1. The process begins by identifying any monosemous strings associated with *c* using the `MRCON` and `AMBIGLUI` tables (shown in line 3). The monosemous strings are then used to identify examples from PubMed (line 4) by creating a query from their disjunction. For example, if $monosemousrels = \{m_1, m_2, \ldots m_{|M|}\}$ the query generated would be $m_1\ OR\ m_2\ OR\ \ldots\ m_{|M|}$. The examples retrieved by this query have to be adapted by replacing the occurrence of the monosemous string with the ambiguous term. The process stops if enough (i.e. *target*) examples have been identified. Otherwise the search continues by using the direct relatives of *c* (line 11) to provide strings that can be used to search PubMed. If further examples are still required then indirect relatives are considered. The *maxlevel* parameter is used to ensure that the relatives being used are not too distantly related to *c*. The maximum value of this parameter was limited to 2 for the experiments described later in the paper.

Algorithm 1. Generate labeled examples using monosemous relatives. The algorithm is provided with a CUI (*c*), number of labeled examples to generate (*target*) and a parameter that controls the maximum level of relative to be used (*maxlevel*). The algorithm examines the relatives of *c* by starting with those directly connected to *c* and gradually considering more distant ones. This is achieved by initialising a variable (*level*) to 0 and incrementing it until it reaches *maxlevel* or the required number of examples have been generated. The value of *target* is decreased as labeled examples are generated. The output of the algorithm is a set of labeled examples (*E*).

```
 1: Initialise variables: $level \leftarrow 0$, $cuiset \leftarrow \{c\}$, $E \leftarrow \emptyset$
 2: while $target > 0$ and $level \leqslant maxlevel$ do
 3: $monosemousrels \leftarrow \{m : m \in cuiset \text{ and } monosemous(m)\}$
 4:   Use $monosemousrels$ to search PubMed and return a
      set of examples, $M$.
 5:   if $|M| \geqslant target$ then
 6:       Choose the first $target$ examples from $M$ and
          append them to $E$
 7:     return $E$
 8:   else
 9:     Append $M$ to $E$
10:     $target \leftarrow target - |M|$
11:     $cuiset \leftarrow \{r : c \in cuiset \text{ and } \exists\, mrrel(c, r)\}$
12:     $level \leftarrow level + 1$
13:   end if
14: end whlie
15: return $E$
```

For example, assume we aim to retrieve 70 examples for CUI C0015677 'Fatty acid glycerol esters', which refers to the 'lipid' meaning[9] of "fat" (and can be contrasted with the 'obese' meaning). Fig. 1 shows some of the CUIs that are related to C0015677. The level 0 search identifies two monosemous relatives ("fatty acid glycerol esters" and "fat preparation") which are used to generate the query `"fatty acid glycerol esters"[TIAB] OR "fat preparation"[TIAB]`. (The `[TIAB]` modifier is used to restrict the search to terms that occur within the title or text of the abstract and not other fields, such as the author, and is necessary to avoid spurious matches.) This query returns 21 abstracts that are processed to substitute the monosemous relatives with the ambiguous term. For example, one abstract has the title "Effects of *fatty acid glycerol esters* on intestinal absorptive and secretory transport of ceftibuten" which becomes "Effects of *fats* on intestinal absorptive and secretory transport of ceftibuten." Since more abstracts are required the search continues to the next level (i.e. level 1). Fig. 1 shows a number of CUIs related to C0015677 at this level. The first to be used is C0682952 'lipids, fatty acids, fats and olis' which is used to generate the query `"lipids, fatty acids, fats and oils"[TIAB]`. This query does not return any abstracts and the search continues to CUI 0015678 'unsaturated fats' which is used to generate the query `"unsaturated fat"[TIAB]`. This query returns over 600 abstracts and, since 21 abstracts have already been identified, only 49 of these are required to meet the target of 70 examples. The first 49 abstracts that were retrieved are used. As before, the term "unsaturated fat" is replaced with "fat". For example in another abstract the phrase "Effect of protein, *unsaturated fat*, and carbohydrate intakes" is altered to "Effect of protein, *fat*, and carbohydrate intakes". On completion this process creates 70 pseudo-labeled sentences containing the word

[9] Defined in SNOMED as "ester of glycerol with fatty acids; generally odorless, colorless, and tasteless if pure; fats are insoluble in water, soluble in most organic solvents; they occur in animal and vegetable tissue."

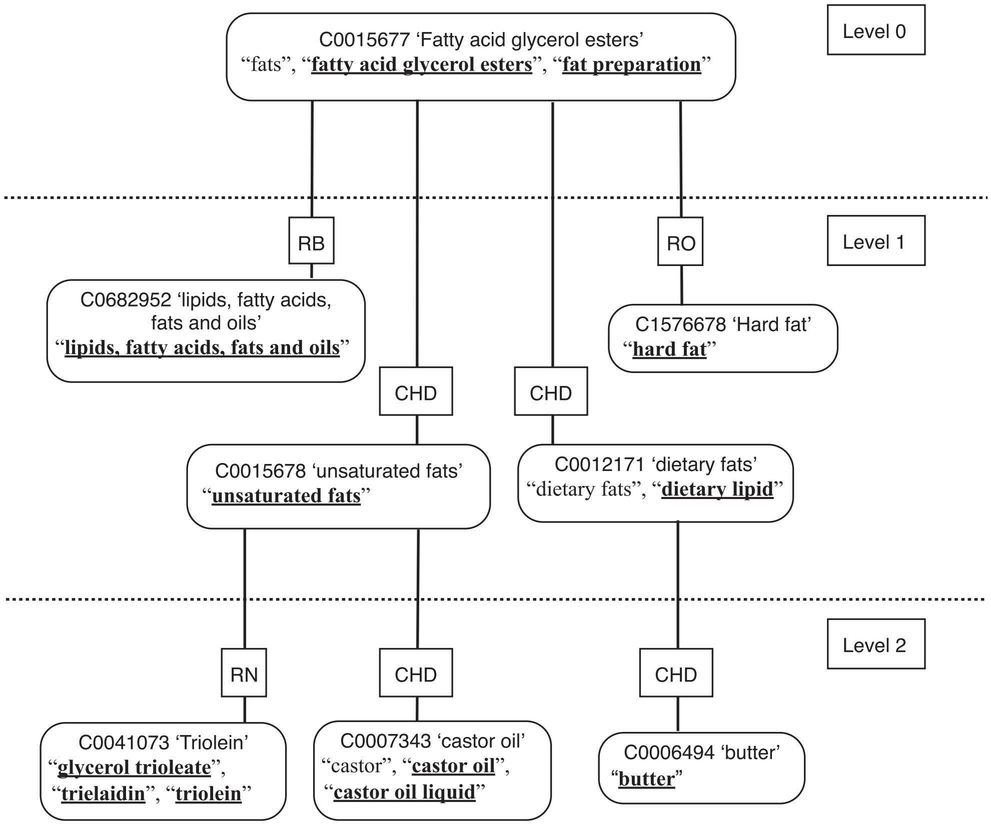
C0015677 'Fatty acid glycerol esters'
"fats", "fatty acid glycerol esters", "fat preparation"
Level 0
RB
RO
Level 1
C0682952 'lipids, fatty acids, fats and oils'
"lipids, fatty acids, fats and oils"
C1576678 'Hard fat'
"hard fat"
CHD
CHD
C0015678 'unsaturated fats'
"unsaturated fats"
C0012171 'dietary fats'
"dietary fats", "dietary lipid"
Level 2
RN
CHD
CHD
C0041073 'Triolein'
"glycerol trioleate", "trielaidin", "triolein"
C0007343 'castor oil'
"castor", "castor oil", "castor oil liquid"
C0006494 'butter'
"butter"

Fig. 1. Diagram showing some of the CUIs involved in the generation of examples for C0015677 'Fatty acid glycerol esters' using the Monosemous Relatives approach. Each CUI is represented as a box including the CUI's identifier, its name and example terms. Relationships between CUIs are shown as lines drawn between them labeled with the type of relation. (For clarity, the diagram does not include all terms when there are a large number of a particular CUI and not all relatives of CUI 0015677 are shown.)

"fat" intended as examples of C0015677. These can then be combined with examples of the other possible meanings of fat and used to train a supervised WSD system.

Fig. 1 shows other monosemous relatives of C0015677 at levels 1 and 2, including "dietary fats" , "hard fats", "glycerol trioleate", "castor oil" and "butter", which would be used by the search if further labeled examples were required.

3.3. Approach 2: co-occurring concepts

The second approach uses co-occurrence information, which has been demonstrated to be valuable for WSD, for example [11,13]. The process is similar to the one that uses monosemous relatives, although co-occurring terms are used. The aim is to identify instances of a term that relate to a particular meaning by creating a search that includes the terms that co-occur with it.

The approach is shown in Algorithm 2, which also aims to generate *target* examples for a particular CUI, *c*. The search starts by identifying the concepts that co-occur with the CUI using information from the `MRCOC` table (line 3). The `MRCOC` table includes information about the frequency of the co-occurrence within Medline and this information is used to identify the 15 most frequently co-occurring concepts (line 4) that are used to identify examples in PubMed (line 5). (The reason for using only the 15 most frequently co-occurring concepts is explained below.) The procedure for generating queries is different from the one that was applied when searching for monosemous relatives. When co-occurring concepts are used the aim is to identify articles that contain the ambiguous term and as many of the co-occurring concepts as possible, since these articles are likely to contain usages of the ambiguous term that refer to *c*. For example, the co-occurring concepts of the CUI C0009443 "Common Cold" include 'rhinovirus', 'influenza' and 'pharmacotherapeutic'. If the word "cold" appears in an article together with these terms then that is a strong indicator that it refers to C0009443.

Appropriate articles are identified by searching PubMed using a series of searches that are designed to identify instances of the ambiguous term that occur with as many co-occurring concepts as possible [28]. The first search in this series looks for articles containing the ambiguous term and all of the co-occurring concepts while subsequent searches are gradually made less restrictive by only requiring fewer co-occurring concepts. For example, assume that for some ambiguous term, *t*, a set of *C* co-occurring concepts have been identified. A query is then formed from the conjunction of all terms ($t \cup C$), i.e. "t *and* c_1 *AND* c_2 *AND* ... $c_{|C|}$". This query is very restrictive and unlikely to match many abstracts so it is made less restrictive by reducing the number of terms in *C* that must be in the abstract. The next search requires the abstract to contain *t* and $|C|-1$ of the terms in *C*, i.e. "t *AND* ((c_1 *AND* c_2 *AND* ... *AND* $c_{|C|-1}$) *OR* (c_1 *AND* c_2 *AND* ... $c_{|C|-2}$ *AND* $c_{|C|}$) *OR* ... *OR* (c_2 *AND* c_3 ... *AND* $c_{|C|}$))". If further abstracts are required the search will be followed by another for abstracts that contain *t* and $|C|-2$ of the terms. This process repeated until either the desired number of abstracts (i.e. *target*) have been retrieved or the search requires *t* and just one of the terms in *C*, i.e. "t *AND* (c_1 *OR* c_2 *OR* .. *OR* $c_{|C|}$)". The maximum number of searches against PubMed that can take place is bounded by the number of co-occurring concepts that are used, i.e. $|C|$. However, for some CUIs the `MRCOC` table defines a huge number of co-occurring concepts which would make this process impractical. Consequently information in the `MRCOC` table about the frequency of the co-occurrence within Medline is used to identify most frequently co-occurring concepts. The top 15 of these are used for the search since we found that this number was sufficient to identify related abstracts without the number of searches becoming unwieldy.

The abstracts returned by these searches contain the ambiguous terms so there is no need to adapt them, as was done when the monosemous relative approach was used.

The remainder of the process is similar to the monosemous relatives approach. If *target* examples have been retrieved the search stops. Otherwise related concepts are identified (line 12) and the process repeated using the concepts that co-occur with them. We also found that expanding the search beyond level 2 relatives led to inappropriate co-occurring terms being included and we did not permit the search to continue any further. This is implemented by limiting the *maxlevel* parameter to 2 for the experiments described later.

Algorithm 2. Generate labeled examples using co-occurring concepts. The algorithm is provided with a CUI (c), number of labeled examples to generate (*target*) and a parameter that controls the maximum level of relative to be used (*maxlevel*). The search is similar to Algorithm 1.

```
 1: Initialise variables: $level \leftarrow 0$, $cuiset \leftarrow \{c\}$, $E \leftarrow \emptyset$
 2: while $target > 0$ and $level \leqslant maxlevel$ do
 3:   $cooccurring \leftarrow \{o : c \in cuiset \text{ and } \exists\, mrcoc(c, o)\}$
 4:   extract top 15 terms from cooccurring and use to create top_cooccurring
 5:   Use top_cooccurring to search PubMed and return a set of examples, $O$.
 6:   if $|O| \geqslant target$ then
 7:     Choose the first target examples from $O$ and append them to $E$
 8:     return $E$
 9:   else
10:     Append $O$ to $E$
11:     $target \leftarrow target - |O|$
12:     $cuiset \leftarrow \{r : c \in cuiset \text{ and } \exists\, mrrel(c, r)\}$
13:     $level \leftarrow level + 1$
14:   end if
15: end whlie
16: return $E$
```

For example, for CUI C0015677 "Fatty acid glycerol esters" the co-occurring terms identified by the level 0 search include "metabolic aspects", "analysis aspect", "chemical aspects", "milk", "mechanism of action qualifier", "oil", "dietary fat", "cattle", "obesity" and "energy metabolism". These terms can be used to search PubMed and retrieve 70 abstracts so there is no need to expand the search by looking for terms that co-occur with this CUI's relatives. Usages of "fat" that are retrieved include "a high *fat* diet" and "CLA-induced milk *fat* depression". The retrieved abstracts can be used by WSD systems as labeled training examples of this sense.

4. Experimental setup

The most direct way to evaluate the pseudo-labeled examples would be to manually examine each and judge whether they are examples of the intended meaning. However, this approach is impractical since the manual judgment of word meaning is a difficult, time consuming process [6,18,30]. Another disadvantage is that any method of evaluation that relies on human judgments is difficult to repeat. To avoid these problems we chose to evaluate the examples by using them as training data for a supervised WSD system. In addition to being convenient and repeatable, this approach also indicates how useful the pseudo-labeled examples are for their intended purpose.

Table 1
Set of terms used for experiments with number of senses, instances and percentage of instances assigned with the most frequent sense (MFS).

Term	Senses	Instances	MFS (%)
Adjustment	3	93	66.67
Blood pressure	3	100	54.00
Culture	2	100	89.00
Evaluation	2	100	50.00
Growth	2	100	63.00
Immunosuppression	2	100	59.00
Implantation	2	98	82.65
Man	3	92	63.04
Mosaic	2	97	53.61
Nutrition	3	89	50.56
Pathology	2	99	85.86
Radiation	2	98	62.24
Repair	2	68	76.47
Sex	3	100	80.00
Ultrasound	2	100	84.00
Variation	2	100	80.00
Weight	2	53	54.72
White	2	90	54.44
	2.28	93.17	67.18

4.1. Data

The NLM-WSD corpus[10] [6] is used for all the experiments described here. This resource has been widely used for experiments on WSD in the biomedical domain, for example [30–33]. The corpus consists of 5000 examples of ambiguous terms found in Medline. It contains 50 ambiguous terms with 100 examples of each. These examples were manually disambiguated by 11 annotators. The guidelines provided to the annotators allowed them to label a senses as "None" if none of the concepts in the UMLS seemed appropriate. These instances could not be mapped onto UMLS and were ignored for our experiments.

The Most Frequent Sense (MFS) [27] is commonly used as a baseline measure in WSD research. It has proved to be a difficult baseline for WSD systems to improve upon, particularly for unsupervised systems [34]. It is computed as the percentage of instances for a particular term that are assigned the most frequently occurring sense in the training data. We excluded terms with high MFS percentage (>90%) from our experiments and also removed those with fewer than 50 instances to ensure that there were a sufficient number of examples to train and evaluate a WSD system.[11] This leaves 18 terms for the experiments described in this paper. These are shown in Table 1 along with the number of senses, instances and MFS percentage for each term. The average MFS percentage of 67.2% is lower than the equivalent figure for the entire NLM-WSD data set (78%), indicating that the terms we use are some of the more challenging ones.

Examples for each of the terms in our data set were generated using the approaches described in Section 3. Previous research [24] has shown that it is important to preserve the relative proportion of senses in the original corpus when automatically retrieving examples. Consequently the number of examples for each sense in the NLM-WSD corpus is used to determine the number of examples we attempt to retrieve for each sense. For instance, there are 44 examples for C1272641 'Systemic arterial pressure', one of the possible CUIs of "blood pressure", so the *target* parameter is set to 44 in Algorithms 1 and 2 when examples of C1272641 are being generated.

4.2. WSD system

Our WSD system has been adapted specifically to disambiguate text from the biomedical domain and has the best reported results

[10] http://wsd.nlm.nih.gov

[11] Other researchers, such as [32,35], also excluded similar terms when using this data set.

over the NLM-WSD corpus [36]. It is based on a system that participated in the Senseval-3 challenge [14] with a performance which was close to the best system for the English and Basque lexical sample tasks. That system was extended by adding extra learning features, including information from domain-specific knowledge sources.

4.2.1. Features

The system uses a wide range of features that can be divided into four categories:

Local collocations: A total of 41 features that represent the context of the ambiguous word. There are two types of local collocation. The first are bigrams and trigrams containing the ambiguous word that are formed from lemmas,[12] word forms or part of speech tag sequences. For example, consider the sentence “A new simple method used to prepare *fat* for injection” in which “fat” is the term being disambiguated. The bigrams and trigrams formed from word forms are “prepare fat”, “fat for”, “to prepare fat”, “prepare fat for” and “fat for injection”.

The second type of collocations are the lemma and word form of content words[13] preceding and following the ambiguous term. For example, the word forms of the nouns preceding and following “fat” in this example are “methods” and “injection”, respectively.

Unigrams: Two types of unigrams are also used as features. Firstly, the lemmas of all content words in the same sentence as the ambiguous word. So the following unigrams would be included for the example sentence “fat”, “injection”, “new”, “method”, “prepare”, “simple” and “use”.

Secondly, the lemmas of all content words in a ±4-word window around the ambiguous term. A list of corpus-specific stopwords was created containing terms that appear frequently in Medline abstracts but which are not useful for disambiguation (e.g. “abstract”, “conclusion”). Any unigrams found in this list were not used as features. The ±4-word window around “fat” in the example sentence is “method used to prepare *fat* for injection” so the following features would be added: “fat”, “injection”, “method”, “prepare” and “use”.

Abstract features: The lemmas of any unigrams that appear at least twice in the entire corpus and are found in the abstract containing the ambiguous terms are added as features. In addition, bigrams in the abstract with high log-likelihood scores [37] are also added. For example, the following features are identified within the abstract containing the example sentence, “clinical”, “experiment”, “study”, “harvesting technique” and “donor site”.

Medical Subject Headings (MeSH): The MeSH terms (see Section 3.1.3) assigned to the abstract in which each ambiguous word occurs are used as features. This feature is the only one that is specific to the biomedical domain. For example, the abstract containing the example phrase has been assigned 14 MeSH terms including “A10.165.114: Adipose Tissue”, “E04.936.664: Transplantation, Autologous” and “M01.060.116: Adult”.

4.2.2. Learning algorithm

Features are combined using the **Vector Space Model**, a memory-based learning algorithm (see [38]), in which each occurrence of an ambiguous word is represented as a vector. These vectors are created by generating the set of features that are assigned to the instances of a particular term and using that as the basis of the vectors that represent all instances of that term. The length of these vectors varies between 3714 (“blood pressure”) and 5093 (“culture”). The average length across all terms is 4282. Each individual occurrence of an ambiguous term is represented as a binary vector in which 1 indicates the presence of a particular feature and 0 its absence. These vectors are sparse; typically fewer than 3% of the positions are non-zero.

During the algorithm’s training phase a single centroid vector, $\vec{C_{s_j}}$, is generated for each possible sense, s_j, using Eq. (1) where T is the set of training examples for a particular term and $sense(\vec{t})$ is the sense associated with the vector $\vec{t}$.

$$\vec{C_{s_j}} = \frac{\sum_{\vec{t_i}\ \epsilon\ T:sense(\vec{t_i})=s_j} \vec{t_i}}{|\vec{t_i}\ \epsilon\ T : sense(\vec{t_i}) = s_j|} \tag{1}$$

Disambiguation is carried out by comparing the vector representing the ambiguous word, $\vec{a}$, against each centroid using the cosine metric, shown in Eq. (2), and choosing the one with the highest score.

$$score(s_j, \vec{a}) = cos(\vec{C_{s_j}}, \vec{a}) = \frac{\vec{C_{s_j}}.\vec{a}}{|\vec{C_{s_j}}||\vec{a}|} \tag{2}$$

4.3. Evaluation

Evaluation is carried out by computing the percentage of examples that are correctly disambiguated, that is when the WSD system assigns the same meaning as the one in the NLM-WSD data set. This is a standard approach for evaluating WSD systems [11,13] that indicates how closely the system output agrees with human judgements.

All experiments use 10-fold cross-validation, a process for estimating system performance by repeatedly testing against different portions of the data. The NLM-WSD data is split into 10 subsets (or “folds”). Each fold is used as test data exactly once with the remaining 9 folds used for training. Results are then averaged across the ten experiments. Pseudo-labeled examples are evaluated by using them as training data and then testing on each of the 10 folds. Pseudocode for the cross-validation process is shown in Algorithm 3.

Algorithm 3. Procedure for carrying out 10-fold cross-validation. The algorithm is provided with a set of data (T) which is split into 10 subsets of equal size. Each subset is used as test data and the overall performance averaged across all 10.

```
Split labeled training data, $T$, into 10 folds ($t_1, t_2, \ldots t_{10}$)
for $i = 1$ to 10 do
    $test\_set \leftarrow t_i$
    if evaluating $pseudo\_labeled\_examples$ then
        $training\_set \leftarrow pseudo\_labeled\_examples$
    else
        $training\_set \leftarrow T - t_i$
    end if
    Train WSD system using $training\_set$, evaluate against $test\_set$ and record performance, $P(i)$.
end for
$performance \leftarrow \frac{\sum_{i=1 to 10} P(i)}{10}$
return $performance$
```

4.4. Combining examples

Our main aim is to generate pseudo-labeled examples that can be used as a substitute for manually labeled ones when none are available. However, previous studies have shown that the performance of WSD systems trained using manually labeled examples can be improved when the training data are augmented with pseudo-labeled examples [24,36].

The most straightforward technique for combining examples is simply to add the pseudo-labeled examples to the manually labeled ones. This is achieved within the 10-fold cross-validation

[12] The lemma of a word is its base form. For example, “child” is the lemma of “children” and “run” is the lemma of “runs”, “ran” and “running”.

[13] Content words are adjectives, adverbs, nouns or adverbs but not words belonging to any other grammatical category.

process (see Section 4.3) by adding all of the automatically generated examples to the training data for each fold and evaluating the WSD system trained using this data against the relevant test fold. This is equivalent to changing line 5 in Algorithm 3 to $training\_set \leftarrow (T - t_i) \cup pseudo\_labeled\_examples$.

One problem with this approach is that the pseudo-labeled examples are simply added to the training data without checking whether they are suitable for training a WSD system. This can be avoided using **nested cross-validation** [39,40], an extension of the standard cross-validation process which allows the simultaneous selection of optimal parameters and estimation of performance. In our case the parameter we wish to optimize is the choice over whether or not to use pseudo-labeled examples as training data for a particular term.

Nested cross-validation essentially introduces a second level of cross-validation which is used to set parameters without ever examining the data that is used for testing. Details are shown in Algorithm 4, which differs from the standard cross-validation approach with the addition of a second cross-validation stage in lines 4 to 21. The purpose of this extra step is to select the best parameter, denoted by α. The values α can take are *none* (when no pseudo-labeled data are used), *mono* (when the examples generated using the monosemous relative approach are included) and *coc* (when the examples generated using co-occurring concepts are applied). If the parameter being tested involves the use of additional data (i.e. has value *mono* or *coc*) then those examples are added to the training data (lines 6 to 8). For each parameter setting 3-fold cross-validation is carried out to estimate performance (lines 9 to 15). The parameter with the best performance is selected (line 17) and used to estimate performance over the test data.

Algorithm 4. Procedure for carrying out nested cross-validation to decide whether additional data should be used. This algorithm extends standard cross-validation (Algorithm 3) by adding an inner loop to estimate performance when different sets of training data are used.

```
1: Split data, $T$, into 10 folds $(t_1, t_2, \dots t_{10})$
2: for $i = 1$ to 10 do
3:   $test\_set \leftarrow t_i$
4:   for all $\alpha \in \{none, mono, coc\}$ do
5:     $training\_set \leftarrow T - t_i$
6:     if $\alpha = mono$ or $\alpha = coc$ then
7:       $training\_set \leftarrow training\_set \cup pseudo\_labeled\_data$
8:     end if
9:     Split $training\_set$ into 3 folds, $v_1$, $v_2$ and $v_3$.
10:     for $k = 1$ to 3 do
11:       $testing\_validation\_set \leftarrow v_k$
12:       $training\_validation\_set \leftarrow training\_set - v_k$
13:       Train WSD system using the $training\_validation\_set$, evaluate on $test\_validation\_set$ and record performance, $O(k)$.
14:     end for
15:     $Q(\alpha) \leftarrow \frac{\sum_{k=1to3} O(k)}{3}$
16:   end for all
17:   Determine best parameter setting, $\alpha^*$, where $\alpha^* = argmax\, Q(\alpha)$
18:   $training\_set \leftarrow T - t_i$
19:   if $\alpha^* = mono$ or $\alpha^* = coc$ then
20:     $training\_set \leftarrow training\_set \cup pseudo\_labeled\_data$
21:   end if
22:   Train the classifier on $training\_set$, test on $test\_set$ and record performance, $P(i)$
23: end for
24: $performance \leftarrow \frac{\sum_{i=1to10} P(i)}{10}$
25: return $performance$
```

An advantage of the nested cross-validation process is that the pseudo-labeled examples are only added to the training data when doing so appears to improve WSD performance. Another advantage is that it can be used to choose between alternative sets of pseudo-labeled examples. For example Algorithm 4 demonstrates how nested cross-validation can be set up to choose between using the pseudo-labeled examples generated using the monosemous relatives approach, those generated using the co-occurring concepts approach or no pseudo-labeled examples. Nested cross-validation can also be adapted for a single set of pseudo-labeled examples. For example, by setting the possible values of the α parameter in Algorithm 4 to *none* and *mono* (i.e. altering line 4 to "**for all** $\alpha \in \{none, mono\}$ **do**"), it could be used to select between the pseudo-labeled examples generated by the monosemous relatives approach and no pseudo-labeled examples. Similarly, setting the same line to "**for all** $\alpha \in \{none, coc\}$ **do**" would select between the examples generated using the co-occurring concepts approach and no pseudo-labeled examples.

5. Results

5.1. Training WSD system using pseudo-labeled examples

Table 2 shows performance of the WSD system when it is trained using the pseudo-labeled examples generated using the monosemous relatives and co-occurring concepts approaches. Examples are generated using three levels of search by setting the *maxlevel* variable in Algorithms 1 and 2 to 0, 1 or 2.

Performance of both approaches improves when the search level increases (see Table 2), although we found that performance starts to decrease when this is increased beyond two. The best performance is obtained using the co-occurring concepts approach when the search level is set to 2, although performance is much lower for when the search level is 0 or 1. Performance of the monosemous relatives approach is not as sensitive to the search level.

5.2. Numbers of examples generated

We found that there were some CUIs for which no examples could be generated, particularly when the search level is 0. There are also CUIs for which the co-occurring concepts approach was unable to generate any examples. For example, there are three possible senses for "blood pressure": C0005823 'Blood pressure (organism function)', C0005824 'Blood pressure determination' and C1272641 'Systemic arterial pressure'. These senses have, respectively, 54, 2 and 44 examples in the NLM-WSD corpus. The co-occurring concepts algorithm identified examples for the first two senses but not the third one since the MRCOC file did not list any concepts for this CUI or its relatives. This lack of training examples is problematic for supervised WSD algorithms since they will never be able to assign senses for which there is no data. In the case of "blood pressure" it will only be able to assign the first two senses and cannot ever assign the third sense, despite the fact that it represents 44% of the test examples from the NLM-WSD corpus. Supervised WSD systems can only assign senses that are included in the training data and in this case there will be no examples of the third sense.

Table 2
WSD performance using automatically acquired examples for various search levels.

	Search level		
	0	1	2
Monosemous relatives	67.74	71.98	71.98
Co-occurring concepts	44.94	68.62	72.36

The number of examples that were obtained at each of the levels of search for the two approaches is shown in Table 3. For each term this table shows the number of examples which we aim to obtain in the column labeled 'target' and the number retrieved by the two approaches for each of the three search levels in the sets of columns labeled '0', '1' and '2'. The search for examples stops for a term if the same number of examples in the subset of the NLM-WSD data set used for evaluation (see Table 1) are found. These figures are printed in underlined bold font. (For clarity we do not print values where the number of examples does not increase in a subsequent search.) For example, we aim to retrieve 100 examples for the term "blood pressure" and using the monosemous relatives approach this number can be identified in the level 1 search. For the two terms where the target number examples could not be found ("blood pressure" and "evaluation") there is no bold figure. The total number of examples retrieved and the number of terms for which the target number of examples had been met is shown in the bottom two rows of Table 3.

More examples are retrieved for the monosemous relative approach than for the co-occurring concepts and this approach generated the target number examples for all terms when the search level is 1. The co-occurring concepts approach only returns examples for 15 of the 18 terms after the same search level.

To further analyse the pseudo-labeled examples we computed the performance for each approach using only the terms for which the target number of examples had been obtained. These results are shown in the rows of Table 4 labeled 'Pseudo-labeled'. The rows labeled 'MFS' and 'Manual' show the MFS percentage and performance using the manually labeled examples from the NLM-WSD data set on the relevant set of terms. The rows labeled 'Terms' show the number of terms for which target number of examples were generated. These results show that performance using the automatically generated examples consistently outperforms the MFS baseline. This indicates that the pseudo-examples do not simply reflect the distribution of examples in the NLM-WSD data set. In general their performance is not as good as using the examples from the NLM-WSD data set but this is to be expected since those examples were manually labeled.

Table 4
Results at each search level over the terms for which enough examples have been generated. Performance using the relevant subset of the NLM-WSD corpus is also included.

	Search level		
	0	1	2
Monosemous relatives			
Terms	11	18	18
Pseudo-labeled	77.36	71.98	71.98
MFS	69.63	67.18	67.18
Manual	84.50	82.60	82.60
Co-occurring concepts			
Terms	3	15	16
Pseudo-labeled	63.00	72.93	73.38
MFS	58.50	68.29	68.18
Manual	57.50	82.89	82.87

The co-occurring concepts approach could only meet the example target for three terms when the search level was set to 0 (see Table 3). Interestingly the automatically generated examples outperform both the MFS baseline and performance using the manually generated examples for these three terms. Performance using the manually labeled examples is actually worse than the MFS percentage by 1%, suggesting that these three terms are particularly difficult to disambiguate.

5.3. Combining examples

The two versions of cross-validation described in Section 4.4 were applied to the entire set of 18 ambiguous terms and the results shown in Table 5. The row labeled "Standard cross-validation" shows the result when the sets of examples are combined using the simple cross-validation process (ie. the pseudo-labeled examples are always added to the manually labeled training data). Results when nested cross-validation is used are shown in the row labeled "Nested cross-validation". In this row the columns headed "Monosemous Relatives" and "Co-occurring Concepts" shows results when nested cross-validation selects between using one of the sets of pseduo-labeled examples or no pseduo-labeled exam-

Table 3
Number of examples generated at a range of search levels using two approaches. Missing figures indicate that no additional examples were generated at that search level. Highlighted figures (**underlined bold**) indicate that the target has been met for a term.

Term	Target	Monosemous relatives		Co-occurring concepts		
		0	1	0	1	2
Adjustment	93	**93**		13	31	**93**
Blood pressure	100	46	**100**	56		
Culture	100	90	**100**	11	**100**	
Evaluation	100	**100**		0	51	80
Growth	100	**100**		**100**		
Immunosuppression	100	**100**		59	**100**	
Implantation	98	**98**		**98**		
Man	92	**92**		**92**		
Mosaic	97	8	**97**	52	**97**	
Nutrition	89	61	**89**	61	**89**	
Pathology	99	6	**99**	14	**99**	
Radiation	98	**98**		37	**98**	
Repair	68	**68**		41	**68**	
Sex	100	**100**		15	**100**	
Ultrasound	100	93	**100**	84	**100**	
Variation	100	**100**		20	**100**	
Weight	53	0	**53**	29	**53**	
White	90	**90**		0	**90**	
Total	1677	1343	1667	782	1422	1513
Target met		11	18	3	15	16

Table 5
WSD performance using a variety of combinations of training examples.

	Monosemous relatives	Co-occurring concepts	Combined
Standard cross-validation	80.54	80.79	–
Nested cross-validation	82.42	83.33	83.18

Table 6
Number of folds in which pseudo-labeled examples are used by the nested cross-validation process.

	Mono	Co-occur	Combined
Adjustment	8	8	9
Blood pressure	1	9	9
Culture	1	10	10
Evaluation	0	0	0
Growth	5	9	9
Immunosuppression	2	1	3
Implantation	8	10	10
Man	0	3	3
Mosaic	3	0	3
Nutrition	8	7	9
Pathology	0	0	0
Radiation	10	0	10
Repair	4	10	10
Sex	6	10	10
Ultrasound	9	5	9
Variation	0	0	0
Weight	1	5	5
White	10	4	10
Average	4.2	5.1	6.6

ples. The rightmost column, headed “Combination”, shows the result when it selects between both sets of pseudo-labeled examples and no pseduo-labeled examples.

Performance using the manually labeled training data is 82.6%. Results when the pseudo-labeled examples are combined with the manually labeled examples using standard cross-validation are lower than this figure, although combining the examples using the standard cross-validation produces better performance than when the pseudo-labeled examples are used alone (see Table 2). The most likely reason is that the pseudo-labeled examples are not accurate enough to improve performance and, instead, introduce noise to the learning algorithm.

Results when examples are combined using nested cross-validation are comparable with those obtained using the manually labeled training data. The best results are obtained using only the examples generated using co-occurring concepts. The result, 83.33%, is a modest improvement over using only the manually labeled data (Wilcoxon Signed Ranks test, $p < 0.1$). Performance using the examples generated using the monosemous relatives approach are not as good. Overall performance is reduced when they are combined with those generated using co-occurring concepts and worse than the manually labeled examples when used alone.

5.4. Analysis of example combinations

The results reported using the nested cross-validation process in Table 5 do not provide information about the proportion of cases in which the pseudo-labeled examples were actually used. Table 6 shows the number of folds in which the additional data were used together with the original data, rather than the original data being used alone, during the nested cross-validation. (The figures in this table refer to the 10 “outer” folds of Algorithm 4 rather than the “inner” folds shown on lines 10 to 14.) This table shows results when nested cross-validation selects between one of the sets of pseudo-labeled examples and the manually labeled data in the “Mono” and “CoOccur” columns, respectively. (These refer to the “Monosemous Relatives” and “Co-occurring Concepts” columns of Table 5.) The “Combined” column show the results when nested cross-validation is used to select between both sets of pseudo-labeled examples and no pseduo-labeled examples. (This refers to the “Combined” column of Table 5.)

When the nested cross-validation process has access to both sets of pseudo-labeled examples (see rightmost column of Table 6) the pseduo-labeled examples are used in every fold for six of the terms and in nine of the 10 folds for another five terms. Table 6 also shows that examples generated using the two approaches are applied to different numbers of folds for each term. For example, for the term “blood pressure” the pseudo-labeled examples generated using the monosemous relative approach are only applied in a single fold while those created using the co-occurring concepts approach are applied in 9 folds. The correlation between the number of folds in which the pseudo-labeled examples are applied for the monosemous relatives and co-occurring concepts approaches is only 0.27 (Pearson’s correlation coefficient). The terms for which the pseudo-labeled examples generated using the monosemous relatives approach were used in many of the folds, for example “radiation” and ”white”, tended to have clear distinctions between the possible senses. These terms also have monosemous relatives that were useful for identifying abstracts containing examples of those senses. For example, the two possible CUIs for “radiation” are C0034519 “Electromagnetic energy” and C0034618 “Radiation Therapy”. The monosemous relatives for C0034519 were “electromagnetic waves”, “electromagnetic energy” and “electromagnetic radiation” while those for C0034618 were “cancer radiotherapy”, “radiotherapy” and “radiotherapeutics”. These relatives were suitable for identifying abstracts containing the term being used to refer to one of the CUIs.

When the pseudo-labeled examples generated using the monosemous relatives approaches were not useful, such as for “evaluation” and “man”, the monosemous relatives extracted from the UMLS for a particular CUI were not able to identify Medline abstracts containing examples of the term that relate to that CUI. A good example of this occurred for the term “pathology” which has two possible CUIs: C0030664 “Occupation or Discipline” and C0677042 “Pathologic Function”. The monosemous relative for the first of these CUIs was “clinical pathology” but this returned documents containing examples of the term “pathology” being used in ways that did not relate to that CUI. For example, one of these abstracts contains the phrase “We present the case of an adult man with a complex clinical pathology”.

The co-occurring concepts approach generated useful pseudo-labeled examples when the co-occurring concepts are clearly distinct. For example, the two possible CUIs for “culture” are C0010453 “Anthropological Culture” and C0430400 “Laboratory culture”. The set of co-occurring concepts for C0010453 include “health attitudes”, “ethnic group”, “cognition”, “questionnaires” and “mental disorder” while those for C0430400 include “polymerase chain reaction”, “bacterial DNA”, “bacteria”, “DNA sequence analysis” and “restriction fragment length polymorphism”. However, the examples generated using this approach are less useful when co-occurring concepts are shared between the possible meanings. For example, the two CUIs for “evaluation” are C0220825 “Evaluation: Intellectual Product or Research Activity” and C0175637 “Health Activity”. The co-occurring concepts for C0175637 include “methodology”, “life quality”, “quality of life” and “standards characteristics” which are also co-occurring concepts for C0220825. (“life quality”, “quality of life” and “standards characteristics” occur during the level 1 search for C0220825 and “methodology” during the level 2 search.) Co-occurring concepts that are shared by the possible meanings are not useful in searches that are designed to identify documents containing one meaning in particular and this leads to the generation of pseudo-labeled examples that are not useful for WSD.

6. Discussion

Results of the example combination experiments (Section 5.3) show that a small improvement can be obtained when pseudo-labeled examples are combined with manually labeled ones. This improvement is modest but should be interpreted in the context of the experimental setup. The approaches described here are compared against the WSD system with the best reported results for the data set that we evaluate against. This system was trained using manually labeled data and represents a high standard of performance. We have also been careful to ensure that we used an appropriate process for combining examples (nested cross-validation) and avoided using one that produces unrealistically high performance [40].

The results also show that it is important to apply the pseudo-labeled examples in an appropriate way. Simply adding them to the manually labeled data does not improve performance but using nested cross-validation does. Combining the examples generated using the co-occurring concepts approach with the manually labeled examples from the NLM-WSD corpus generated a small improvement in performance for a system that has reported the best results for this data set. There is no such improvement when the examples generated using the monosemous relative approach is used, suggesting that examples generated with the co-occurring concepts are more useful.

Results when the pseudo-labeled examples are used alone (Section 5.1) are consistent with those previously reported [1] for an approach that is very similar to the monosemous relative approach with the search level set to 0. They reported that this approach achieved high performance but could only be applied to around half the ambiguous terms. We also found that this approach could generate sufficient examples for half the terms and also that expanding the search to include indirectly related concepts allows examples to be generated for all terms (see Section 5.2).

7. Conclusions

This paper describes two approaches for pseudo-labeling training data for WSD systems using information in the UMLS Metathesaurus. Both approaches use this information to create search queries that aim to identify Medline abstracts containing examples used with the relevant meaning. The first approach, monosemous relatives, uses information about related concepts and their ambiguity while the second, co-occurring concepts, also uses information about related concepts but in combination with information about the concepts that tend to co-occur with them.

The pseudo-labeled examples generated by these approaches consistently outperformed the MFS baseline when they were used as training data for a supervised WSD system. Combining the examples generated using co-occurring concepts with manually labeled data leads to a modest improvement in WSD performance. These methods allow examples to be generated without the need for manual annotation and help to avoid this significant bottleneck in the application of WSD.

Acknowledgments

We are grateful to the two anonymous reviewers for their detailed feedback and helpful suggestions. We also thank Bridget McInnes for providing the mapping from the NLM-WSD corpus to UMLS and David Martinez for his assistance with the WSD system. This work was supported by the UK Engineering and Physical Sciences Research Council (Grants EP/D069548/1 and EP/E004350/1).

References

[1] Liu H, Johnson S, Friedman C. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. J Am Med Inform Assoc 2002;9(6):621–36.

[2] Hunter L, Cohen K. Biomedical language processing: what's beyond PubMed? Mol Cell 2006;21:589–94.

[3] Westbrook J, Coiera E, Gosling A. Do online information retrieval systems help experienced clinicians answer clinical questions? J Am Med Inform Assoc 2005;12:315–21.

[4] Lau A, Coiera E. Impact of web searching and social feedback on consumer decision making: a prospective online experiment. J Med Internet Res 2008;10(1):e2.

[5] Humphreys L, Lindberg D, Schoolman H, Barnett G. The Unified Medical Language System: an informatics research collaboration. J Am Med Inform Assoc 1998;1(5):1–11.

[6] Weeber M, Mork J, Aronson A. Developing a test collection for biomedical word sense disambiguation. In: Proceedings of AMIA symposium. Washington, DC; 2001. p. 746–50.

[7] Aronson A, Bodenreider O, Chang H, Humphrey S, Mork J, Nelson S, et al. The NLM indexing initiative. In: Proceedings of the AMIA symposium. Los Angeles, CA; 2000. p. 17–21.

[8] Friedman C. A broad coverage natural language processing system. In: Proceedings of the AMIA symposium. Los Angeles, CA; 2000. p. 270–4.

[9] Weeber M, Klein H, Aronson A. Text-based discovery in biomedicine: the architecture of the DAD-system. In: Proceedings of the AMIA symposium. Los Angeles, CA; 2000. p. 903–7.

[10] Swanson D, Smalheiser N. An interactive system for finding complementary literatures: a stimulus to scientific discovery. Artif Intell 1997;91:183–203.

[11] Ide N, Véronis J. Introduction to the special issue on word sense disambiguation: the state of the art. Comput Linguist 1998;24(1):1–40.

[12] Agirre E, Edmonds P, editors. Word sense disambiguation: algorithms and applications. Text, speech and language technology. Springer; 2007.

[13] Navigli R. Word sense disambiguation: a survey. ACM Comput Surv 2009;41(2):1–69.

[14] Mihalcea R, Chklovski T, Kilgarriff A. The Senseval-3 English lexical sample task. In: Proceedings of Senseval-3: the third international workshop on the evaluation of systems for the semantic analysis of text. Barcelona, Spain; 2004. p. 25–8.

[15] Mooney R. Comparative experiments on disambiguating word senses: an illustration of the role of bias in machine learning. In: Proceedings of the conference on empirical methods in natural language processing. Philadelphia, PA; 1996. p. 82–91.

[16] Ng H. Getting serious about word sense disambiguation. In: Proceedings of the SIGLEX workshop "Tagging Text with Lexical Semantics: What, why and how?". Washington, DC; 1997. p. 1–7.

[17] Aronson A. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the AMIA symposium; 2001. p. 17–21.

[18] Artstein R, Poesio M. Inter-coder agreement for computational linguistics. Comput Linguist 2008;34(4):555–96.

[19] Brody S, Lapata M. Good neighbors make good senses: exploiting distributional similarity for unsupervised WSD. In: Proceedings of the 22nd international conference on computational linguistics (Coling 2008). Manchester, UK; 2008. p. 65–72.

[20] Gale W, Church K, Yarowsky D. A method for disambiguating word senses in a large corpus. Comput Humanit 1993;26:415–39.

[21] Ng H, Wang B, Chan S. Exploiting parallel texts for word sense disambiguation: an empirical study. In: Proceedings of the 41st annual meeting of the association for computational linguistics (ACL-03). Sapporo, Japan; 2003. p. 455–62.

[22] Leacock C, Chodorow M, Miller G. Using corpus statistics and word-net relations for sense identification. Comput Linguist 1998;24(1):147–65.

[23] Fellbaum C, Grabowski J, Landes S, Baumann A. Matching words to senses in WordNet: naive vs. expert differentiation of senses. In: Fellbaum C, editor. WordNet: an electronic lexical database and some applications. Cambridge, MA: MIT Press; 1998.

[24] Agirre E, Martínez D. Unsupervised WSD based on automatically retrieved examples: the importance of bias. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP-04). Barcelona, Spain; 2004.

[25] Stevenson M, Guo Y, Alamri A, Gaizauskas R. Disambiguation of biomedical abbreviations. In: Proceedings of the BioNLP 2009 workshop. Boulder, Colorado; 2009. p. 71–9.

[26] Lin D. Automatic retrieval and clustering of similar words. In: Proceedings of the 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics, vol. 2. Montreal, Quebec, Canada; 1998. p. 768–74.

[27] McCarthy D, Koeling R,Weeds J, Carroll J. Finding PredominantWord senses in untagged text. In: Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-2004). Barcelona, Spain; 2004. p. 280–87.

[28] Stevenson M, Guo Y, Gaizauskas R. Acquiring sense tagged examples using relevance feedback. In: Proceedings of the 22nd international conference

on computational linguistics (Coling 2008). Manchester, UK; 2008. p. 809–16.

[29] Nelson S, Powell T, Humphreys B. The Unified Medical Language System (UMLS) project. In: Kent A, Hall CM, editors. Encyclopedia of library and information science. Marcel Dekker Inc.; 2002.

[30] Savova GK, Coden A, Sominsky IL, Johnson R, Ogren PV, de Groen PC, et al. Word sense disambiguation across two domains: biomedical literature and clinical notes. J Biomed Inform 2008;41(6):1088–100.

[31] Joshi M, Pedersen T, Maclin R. A comparative study of support vector machines applied to the word sense disambiguation problem for the medical domain. In: Proceedings of the second Indian conference on artificial intelligence (IICAI-05). Pune, India; 2005. p. 3449–68.

[32] Leroy G, Rindflesch T. Effects of information and machine learning algorithms on word sense disambiguation with small datasets. Int J Med Inform 2005;74(7-8):573–85.

[33] McInnes B, Pedersen T, Carlis J. Using UMLS concept unique identifiers (CUIs) for word sense disambiguation in the biomedical domain. In: Proceedings of the annual symposium of the American medical informatics association. Chicago, IL; 2007. p. 533–7.

[34] Pradhan S, Loper E, Dligach D, Palmer M. SemEval-2007 Task-17: English lexical sample, SRL and all words. In: Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007). Prague, Czech Republic; 2007. p. 87–92.

[35] Humphrey S, Rogers W, Kilicoglu H, Demner-Fushman D, Rindflesch T. Word sense disambiguation by selecting the best semantic type based on journal descriptor indexing: preliminary experiment. J Am Soc Inform Sci Technol 2006;57(5):96–113.

[36] Stevenson M, Guo Y, Gaizauskas R, Martinez D. Knowledge sources for word sense disambiguation of biomedical text. In: Proceedings of the workshop on current trends in biomedical natural language processing at ACL 2008. Columbus, OH; 2008. p. 80–7.

[37] Pedersen T. A decision tree of bigrams is an accurate predictor of word sense. In: Proceedings of the second meeting of the north American chapter of the association for computational linguistics (NAACL-01). Pittsburgh, PA.; 2001. p. 79–86.

[38] Agirre E, Martínez D. The Basque Country University system: English and Basque tasks. In: Mihalcea R, Edmonds P, editors. Senseval-3: third international workshop on the evaluation of systems for the semantic analysis of text. Barcelona, Spain; 2004. p. 44–8.

[39] Scheffer T. Error estimation and model selection. Technischen Universität Berlin, School of Computer Science; 1999.

[40] Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF. GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. Int J Med Inform 2005;74(7-8):491–503.