Lang Resources & Evaluation
DOI 10.1007/s10579-012-9180-5

ORIGINAL PAPER

Creating a system for lexical substitutions from scratch using crowdsourcing

Chris Biemann

Abstract This article describes the creation and application of the Turk Bootstrap Word Sense Inventory for 397 frequent nouns, which is a publicly available resource for lexical substitution. This resource was acquired using Amazon Mechanical Turk. In a bootstrapping process with massive collaborative input, substitutions for target words in context are elicited and clustered by sense; then, more contexts are collected. Contexts that cannot be assigned to a current target word's sense inventory re-enter the bootstrapping loop and get a supply of substitutions. This process yields a sense inventory with its granularity determined by substitutions as opposed to psychologically motivated concepts. It comes with a large number of sense-annotated target word contexts. Evaluation on data quality shows that the process is robust against noise from the crowd, produces a less fine-grained inventory than WordNet and provides a rich body of high precision substitution data at low cost. Using the data to train a system for lexical substitutions, we show that amount and quality of the data is sufficient for producing high quality substitutions automatically. In this system, co-occurrence cluster features are employed as a means to cheaply model topicality.

Keywords Amazon Turk · Lexical substitution · Word sense disambiguation · Language resource creation · Crowdsourcing

This work was done while the author worked for the Powerset division of Microsoft Bing.

C. Biemann (✉)
Technische Universität Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany
e-mail: biem@cs.tu-darmstadt.de
URL: http://www.ukp.tu-darmstadt.de/

Published online: 30 March 2012

1 Introduction

This research is motivated by a problem in semantic information retrieval (SemIR): How can a query be matched to a relevant document in absence of keyword overlap? A naive synonym expansion—be it on the query side or on the index side—fails for words with multiple meanings in absence of word sense disambiguation (WSD). In this introduction, we re-examine the role of WSD in IR under the aspect of semantic matching. Then we describe how the use of a crowdsourcing platform like Amazon Mechanical Turk (AMT) for the targeted problem of lexical substitution can overcome these shortcomings.

1.1 Word sense disambiguation

As an enabling step for semantic applications, word sense disambiguation (WSD) is the task of assigning word senses for ambiguous words in context. In the supervised setting, a sense-labeled training corpus is used to train a model for each target word. This model is used to classify the occurrence of a target word in an unseen context into one of the senses that occurred in the training. In the unsupervised or knowledge-based setting, a semantic resource (like WordNet) is employed to identify senses in context by merely using the semantic resource itself, not the training examples.

In particular for semantic search with matching beyond keywords, one is interested in the possible substitutions for a word in context to be able to expand the index with these. In case of ambiguous words, it is necessary to identify the correct sense first to avoid spurious expansions that lead to mismatches.

A system for lexical substitution thus can be realized through a WSD system that assigns word senses in context and that is equipped with a set of substitutions per sense. Notice that when interested merely in lexical substitution quality, the sense inventory itself plays only an intermediate role as we are not interested in identifying one of possible senses given by the inventory, but in a set of acceptable substitutions. In particular, we do not need to map the inventory used for this to other inventories like WordNet.

Disappointing progress in Word Sense Disambiguation (WSD) competitions has often been attributed to problems with WordNet (Miller et al. 1990). While a valuable resource to the NLP community, WordNet was not originally designed for WSD. Still being the best option available in terms of cost and coverage, it became the standard resource in Senseval and Semeval competitions, which sparked a variety of approaches to WSD. In general, supervised approaches have reached higher levels of accuracy, but are hampered by the acquisition bottleneck in creating labeled training data for word senses. WordNet has widely been used as a word sense inventory.

High WSD performance scores using WordNet suffer from the extremely fine-grained distinctions that characterize the resource and by the relatively little available data for senses in contexts (cf. e.g. Agirre and Edmonds 2006). For example, of the eight noun senses of “hook”, four refer to a bent, curvy object.

However, in the entire SemCor (Mihalcea 1998) there is only one occurrence recorded for this sense altogether, so for most of the senses the only data available are the glosses and the relations to other synsets. Even if some fine-grained classes are combined by clustering WordNet senses (Mihalcea and Moldovan 2001), alignment of the sense inventory and the target domain or application remains a problem. Using WordNet, or any predefined inventory, for WSD may result in a mismatch with the target domain of the application. If it does not fit well, domain adaptation will be required—a costly endeavor that will likely have to be repeated. Corpus-based word sense acquisition, on the other hand, guarantees a match between inventory and target domain.

Since semantic annotation tasks are notoriously difficult and low inter-annotator agreement goes hand in hand with low WSD scores, the OntoNotes project (Hovy et al. 2006) aimed at high agreement through word sense grouping to mitigate the problems above. In Hovy et al. (2006) it was shown that enforcing more than 90 % inter-annotator agreement and more coarse-grained sense groupings in fact could ensure accuracy levels close to 90 %. These groupings were employed in the lexical sample task of SemEval 2007 (Pradhan et al. 2007), which we use for evaluation in Sect. 4.

However, manual methods of sense reduction such as those used in OntoNotes are costly and may not be scalable due to their dependence on highly trained annotators working for extended periods to learn the annotation protocols.

1.2 The case of WSD for SemIR

A major potential application of WSD is to serve as a mechanism to supply correct substitutions in context for ambiguous words: After WSD has resolved the correct sense, the set of synonyms is restricted to the ones compatible with this sense and probably restricted further by other techniques. The ability to make the right substitutions, in turn, gives rise to fuzzy semantic matching in Information Retrieval.

The intuition that WSD should benefit traditional IR has sparked a number of attempts to leverage disambiguation techniques for retrieval, see Agirre and Edmonds (2006, Chap. 11.3) for a summary. Results have been mixed at best. This is commonly attributed to two reasons. The first reason is seen in insufficient accuracy of current WSD systems: As Sanderson (1994) estimated, at least 90 % accuracy is required before the benefits of WSD-supplied substitutions or term expansions outweigh the drawbacks from errors. Somewhat contrary, Stokoe (2005) suggests that an accuracy of 55–76 % should be sufficient, which is more in the range of what fine-grained WSD systems perform at. Both studies employ pseudo words to simulate ambiguity, which is somewhat artificial. Second, and more important, the headroom of WSD in keyword search is smaller than intuitively expected, due to what Sanderson (2000) calls the "query collocation effect": The longer the query, the more the query words disambiguate each other. Figure 1a illustrates the query collocation effect: Whereas the ambiguous query "case" retrieves all documents containing "case", longer queries like "court case" or "plastic case" restrict the pool of relevant documents to the ones where "case" (mostly) is used in the desired sense, without an explicit disambiguation step. This

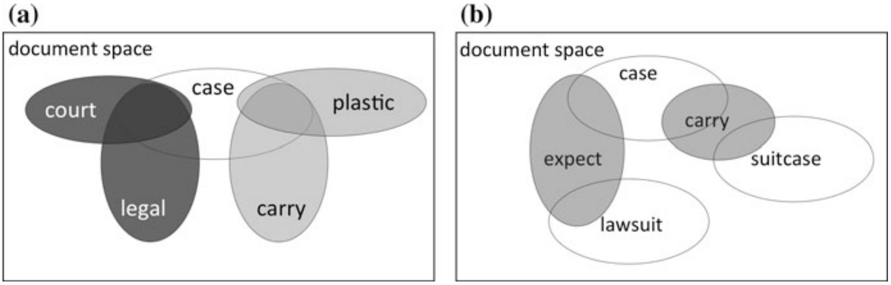


Fig. 1 **a** Query word collocation effect for keyword-based IR, **b** absence of mutual disambiguation in synonym expansion without disambiguation

changes when performing synonym expansion as exemplified in Fig. 1b for the query “carry case”. The keyword “case” is expanded to both “suitcase” and “lawsuit”, the keyword “carry” is expanded to “expect” (via “to be pregnant with”), resulting in a match for documents that match the keywords “expect lawsuit”. Note that this effect will not be alleviated by one-off-fixing the synonym inventory: Especially highly frequent words are ambiguous, and restricting oneself to only unambiguous words for expansion will most likely impact fewer queries than the cost of getting at an appropriate synonym list is worth.

When moving from keyword-based IR to more semantic, fuzzy matching approaches, it is apparent that a simple synonym list will not do the trick because of semantic ambiguity. As observed in the previous section, it is hardly feasible to use existing, general-purpose semantic resources like WordNet for disambiguation and it is expensive to tailor them to one’s needs. For the given task, we are actually much less interested in assigning senses from a predefined inventory to ambiguous words in context; what we require to perform our fuzzy matching is a method that tells us alternative ways of expressing the same meaning. For the purpose of this paper, we restrict ourselves to single-word substitutions in context. Note that we do neither require a taxonomy structure nor do we have to group words into concepts or synsets.

1.3 Mechanical Turk to the rescue

In this work, we pursue an alternative path to expert annotators: we set up a bootstrapping process that relies on redundant human input to crystallize the resource we need for training a supervised lexical substitution system. Instructions are kept short and tasks are kept simple. Because some degree of noisy input is tolerated in this method, naïve annotators can be used, and data can be produced quickly, at low cost and without access to a full in-house annotation staff. Also, we tailor the resource closely to our task, preferring a specialized solution rather than aiming at a general solution that fits our task only somewhat.

As a crowdsourcing platform, we use Amazon Mechanical Turk (AMT). AMT allows requesters to post arbitrary tasks or HITs (Human Intelligence Tasks), which are done for pay by a large pool of annotators. AMT allows to specify the number of annotators per task item, as well as to restrict the annotator pool by various criteria. The quality of annotations from AMT has been shown to be comparable to professional annotators when answers from four or more different annotators are combined, see Snow et al. (2008). This, however, only applies to comparably simple tasks due to the nature of crowdsourcing: the more complicated the task, the more training overhead is involved for the annotator, who in turn steers away to tasks promising a more immediate reward. Also, the necessity to combine the work of several annotators makes it necessary to design the task in a way that a large overlap between the answers of several people on the same task is likely.

1.4 Outline

The remainder of this paper is organized as follows. In Sect. 2, we describe the design rationale for creating the TWSI using three different kinds of HITs. Section 3 describes the TWSI version 1.0 quantitatively and qualitatively. In Sect. 4, we describe how the TWSI was used to build a system for lexical substitution based on a WSD system. The WSD step and substitution quality is evaluated, showing high precision. Finally, Sect. 5 concludes.

2 Creating a lexical substitution resource with Amazon Turk

This section describes in detail how we employed three turker tasks (HITs) to elicit our semantic resource. After presenting the tasks in isolation, we describe the process that connects them and exemplify the data flow between components.

2.1 Three turker tasks

This section is devoted to three elementary tasks given to annotators whom we will refer to as *turkers* in the AMT platform. The nature of crowdsourcing makes it necessary to follow some guidelines when designing tasks: (1) Both tasks and instruction sets for those tasks must be simple to hold training to a minimum, (2) redundancy is necessary to assure quality. The inherent noisiness of the process requires that only answers supplied multiple times by different turkers should be accepted.

Requiring redundancy in answers is also important in identifying deliberate scammers.

2.1.1 Task 1: Finding substitutions

The rationale behind this task is to be able to identify word senses by the differences in possible substitutions. For information retrieval applications, we find this substitution-based definition of senses desirable. Here, WSD is input for determining which lexical expansions should be used for matching, so the concept of substitutability is central.

In this task, turkers are presented with a sentence containing a target word emphasized in bold. They are asked to supply possible substitutions for the bolded word in the specific sentential context. Turkers must supply at least one, and up to a maximum of five substitutions. While not specifically instructed, turkers used multiword as substitutions when they found this appropriate. When the input covered several words, turkers were instructed to identify the span: e.g. for the target “station” in a sentence like “they arrived at the railroad station”, turkers could supply “railroad station” along with other substitutes.

In addition, turkers are asked to state whether the sentence is a good or acceptable representative of the target word meaning. When turkers indicate that assigning a sense to the target is hard or impossible, part-of-speech violations and insufficient contexts are potential culprits. Figure 2 shows an exemplary HIT as seen by the turker.

2.1.2 Task 2: Aligning senses

This task measures how similar the senses of two usages of the same word are, as exemplified in Fig. 3. The rationale behind this task is to measure closeness of senses and to be able to merge senses in cases where they are identical. Turkers are shown a pair of sentences that contain the same emphasized target word. They are then asked whether the meaning of this target word in the two sentences is identical, similar, different or impossible to determine. This is very similar to the usage experiments described in Erk et al. (2009), who use a five-point scale to annotate meaning similarity of usages.

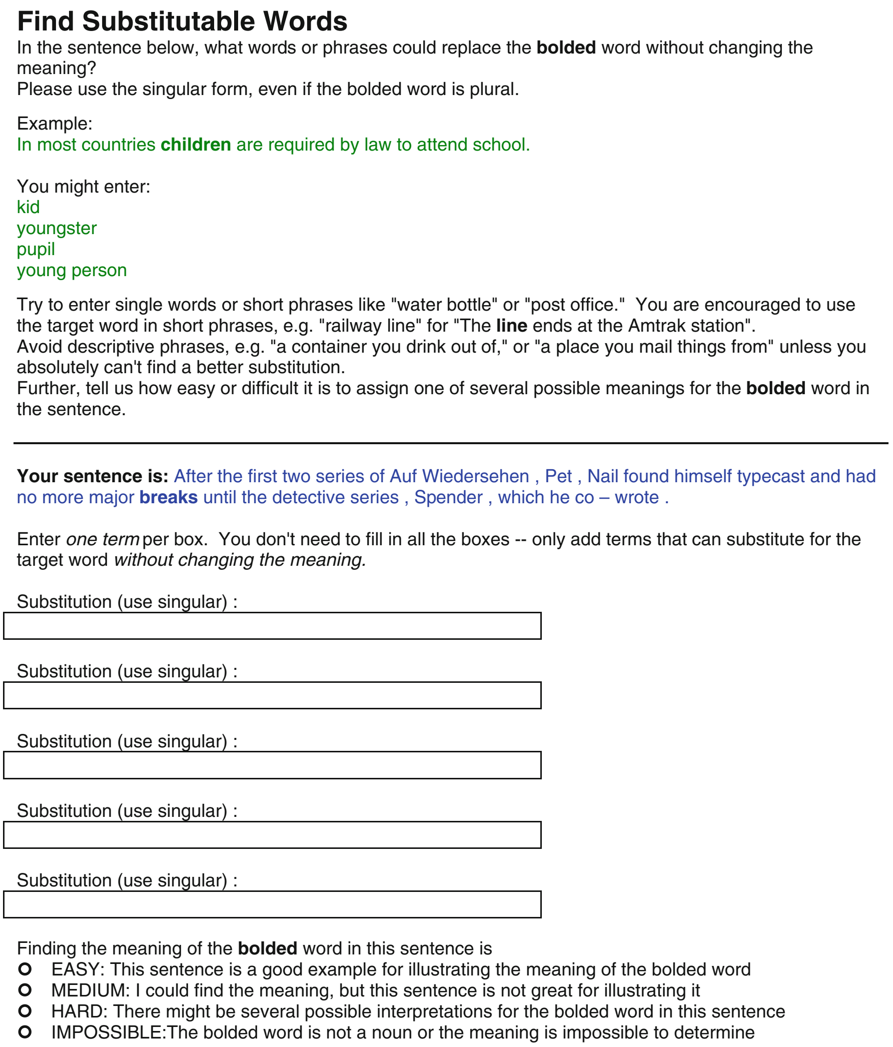
2.1.3 Task 3: Match the meaning

This task presents a sentence with a bolded target word and requires turkers to align usages of the target word to a given inventory: they are asked to choose one of a set of sentences containing the target word in different meanings representing the current sense inventory, see Fig. 4. They also can state that the sense is not covered by the current inventory, or label the sense as impossible to determine. To make the task more efficient and easier, this assignment contains ten of these questions for the same word using the same inventory, so the turker has to understand the inventory choice only once.

2.2 Bootstrapping the word sense inventory

This section describes how a word sense inventory is constructed using the tasks in the previous subsection. Figure 5 provides a schematic overview of how the three tasks are executed in sequence. Note that each target word is processed separately, and that the process is described for a single target word. We will use the noun target “station” for exemplifying the steps.

When using untrained turkers, the process that distills their raw responses to usable data must be formulated in a way that uses redundant answers and is robust



Find Substitutable Words

In the sentence below, what words or phrases could replace the **bolded** word without changing the meaning?
Please use the singular form, even if the bolded word is plural.

Example:
In most countries **children** are required by law to attend school.

You might enter:
kid
youngster
pupil
young person

Try to enter single words or short phrases like "water bottle" or "post office." You are encouraged to use the target word in short phrases, e.g. "railway line" for "The **line** ends at the Amtrak station".
Avoid descriptive phrases, e.g. "a container you drink out of," or "a place you mail things from" unless you absolutely can't find a better substitution.
Further, tell us how easy or difficult it is to assign one of several possible meanings for the **bolded** word in the sentence.

Your sentence is: After the first two series of Auf Wiedersehen , Pet , Nail found himself typecast and had no more major **breaks** until the detective series , Spender , which he co – wrote .

Enter *one term* per box. You don't need to fill in all the boxes -- only add terms that can substitute for the target word *without changing the meaning.*

Substitution (use singular) :

Substitution (use singular) :

Substitution (use singular) :

Substitution (use singular) :

Substitution (use singular) :

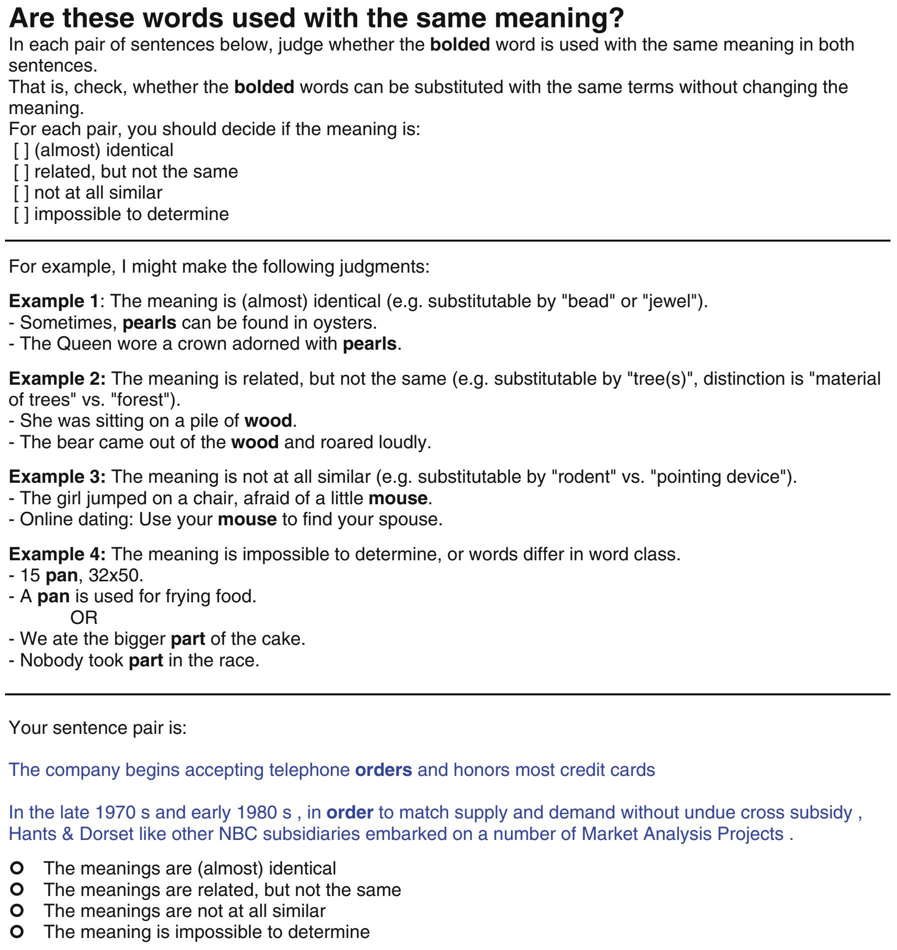
Finding the meaning of the **bolded** word in this sentence is
○ EASY: This sentence is a good example for illustrating the meaning of the bolded word
○ MEDIUM: I could find the meaning, but this sentence is not great for illustrating it
○ HARD: There might be several possible interpretations for the bolded word in this sentence
○ IMPOSSIBLE:The bolded word is not a noun or the meaning is impossible to determine

Fig. 2 HIT with instructions for Task 1: Find Substitutable words. Turkers must supply at least one substitution

against noise. Various thresholds described below have been introduced for these reasons. For trained, professional annotators, most of these constraints could be relaxed.

2.2.1 From start to Task 1

For the target word, 15 random sentences are selected from a corpus resource. While 15 sentences is a somewhat arbitrary number, we have found that it ensures that the major senses of the target word are usually covered. These sentences are seeds to initiate the cycle, and need not cover all the senses in the corpus or final inventory.



Are these words used with the same meaning?

In each pair of sentences below, judge whether the **bolded** word is used with the same meaning in both sentences.
That is, check, whether the **bolded** words can be substituted with the same terms without changing the meaning.
For each pair, you should decide if the meaning is:
[ ] (almost) identical
[ ] related, but not the same
[ ] not at all similar
[ ] impossible to determine

For example, I might make the following judgments:

Example 1: The meaning is (almost) identical (e.g. substitutable by "bead" or "jewel").
- Sometimes, **pearls** can be found in oysters.
- The Queen wore a crown adorned with **pearls**.

Example 2: The meaning is related, but not the same (e.g. substitutable by "tree(s)", distinction is "material of trees" vs. "forest").
- She was sitting on a pile of **wood**.
- The bear came out of the **wood** and roared loudly.

Example 3: The meaning is not at all similar (e.g. substitutable by "rodent" vs. "pointing device").
- The girl jumped on a chair, afraid of a little **mouse**.
- Online dating: Use your **mouse** to find your spouse.

Example 4: The meaning is impossible to determine, or words differ in word class.
- 15 **pan**, 32x50.
- A **pan** is used for frying food.
OR
- We ate the bigger **part** of the cake.
- Nobody took **part** in the race.

Your sentence pair is:

The company begins accepting telephone **orders** and honors most credit cards

In the late 1970 s and early 1980 s , in **order** to match supply and demand without undue cross subsidy , Hants & Dorset like other NBC subsidiaries embarked on a number of Market Analysis Projects .

○ The meanings are (almost) identical
○ The meanings are related, but not the same
○ The meanings are not at all similar
○ The meaning is impossible to determine

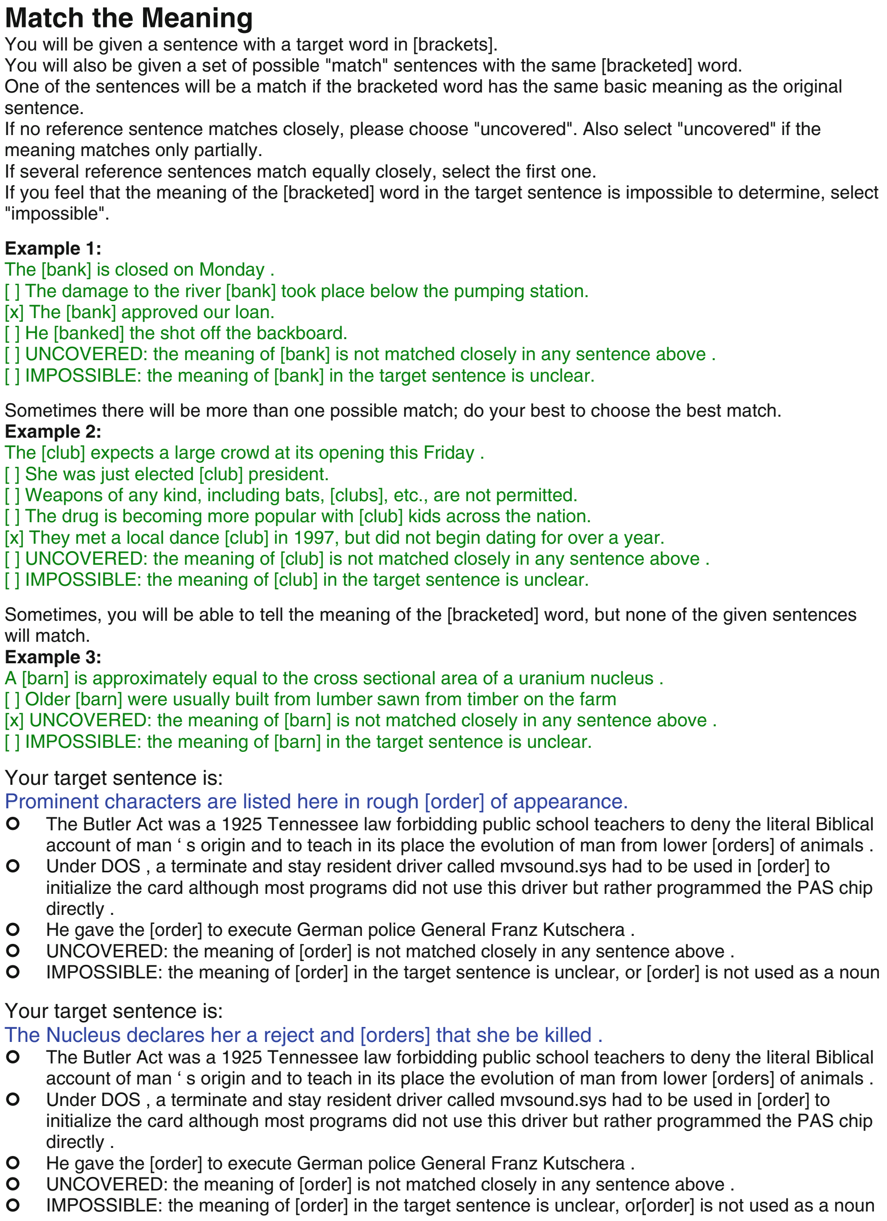
Fig. 3 HIT with instructions for Task 2: Are these words used with the same meaning?

In our experiments, we select by lemma and part-of-speech, so e.g. for the noun target "station" we would select sentences where "station" or "stations" were tagged as a noun. These sentences are presented in Task 1. We will use these three sentences as an example:

- A: The train left the **station**.
- B: This radio **station** broadcasts news at five.
- C: Five miles from the **station**, the railway tracks end.

These were assigned the following substitutions (multiplicity in parentheses):

- A: terminal(5), railway station(3), rail facility(1), stop(1)
- B: radio station(4), network(3), channel(2)
- C: terminal(3), stop(2), train depot(2), railway station(1)



Match the Meaning

You will be given a sentence with a target word in [brackets].
You will also be given a set of possible "match" sentences with the same [bracketed] word.
One of the sentences will be a match if the bracketed word has the same basic meaning as the original sentence.
If no reference sentence matches closely, please choose "uncovered". Also select "uncovered" if the meaning matches only partially.
If several reference sentences match equally closely, select the first one.
If you feel that the meaning of the [bracketed] word in the target sentence is impossible to determine, select "impossible".

Example 1:
The [bank] is closed on Monday .
[ ] The damage to the river [bank] took place below the pumping station.
[x] The [bank] approved our loan.
[ ] He [banked] the shot off the backboard.
[ ] UNCOVERED: the meaning of [bank] is not matched closely in any sentence above .
[ ] IMPOSSIBLE: the meaning of [bank] in the target sentence is unclear.

Sometimes there will be more than one possible match; do your best to choose the best match.
Example 2:
The [club] expects a large crowd at its opening this Friday .
[ ] She was just elected [club] president.
[ ] Weapons of any kind, including bats, [clubs], etc., are not permitted.
[ ] The drug is becoming more popular with [club] kids across the nation.
[x] They met a local dance [club] in 1997, but did not begin dating for over a year.
[ ] UNCOVERED: the meaning of [club] is not matched closely in any sentence above .
[ ] IMPOSSIBLE: the meaning of [club] in the target sentence is unclear.

Sometimes, you will be able to tell the meaning of the [bracketed] word, but none of the given sentences will match.
Example 3:
A [barn] is approximately equal to the cross sectional area of a uranium nucleus .
[ ] Older [barn] were usually built from lumber sawn from timber on the farm
[x] UNCOVERED: the meaning of [barn] is not matched closely in any sentence above .
[ ] IMPOSSIBLE: the meaning of [barn] in the target sentence is unclear.

Your target sentence is:
Prominent characters are listed here in rough [order] of appearance.

○ The Butler Act was a 1925 Tennessee law forbidding public school teachers to deny the literal Biblical account of man ‘ s origin and to teach in its place the evolution of man from lower [orders] of animals .
○ Under DOS , a terminate and stay resident driver called mvsound.sys had to be used in [order] to initialize the card although most programs did not use this driver but rather programmed the PAS chip directly .
○ He gave the [order] to execute German police General Franz Kutschera .
○ UNCOVERED: the meaning of [order] is not matched closely in any sentence above .
○ IMPOSSIBLE: the meaning of [order] in the target sentence is unclear, or [order] is not used as a noun

Your target sentence is:
The Nucleus declares her a reject and [orders] that she be killed .

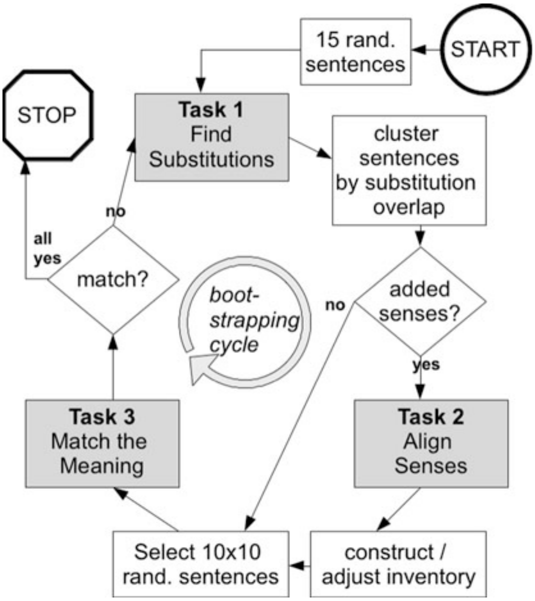
○ The Butler Act was a 1925 Tennessee law forbidding public school teachers to deny the literal Biblical account of man ‘ s origin and to teach in its place the evolution of man from lower [orders] of animals .
○ Under DOS , a terminate and stay resident driver called mvsound.sys had to be used in [order] to initialize the card although most programs did not use this driver but rather programmed the PAS chip directly .
○ He gave the [order] to execute German police General Franz Kutschera .
○ UNCOVERED: the meaning of [order] is not matched closely in any sentence above .
○ IMPOSSIBLE: the meaning of [order] in the target sentence is unclear, or[order] is not used as a noun

Fig. 4 HIT with instructions for Task 3: Match the Meaning. Only two of ten target sentences are shown

2.2.2 *From Task 1 to Task 2*

Having obtained a set of substitutions from the crowd, we compute a weighted similarity graph with the sentences as nodes. Edge weights are given by the amount of overlap in the substitutions of sentences. If two sentences share at least three

Fig. 5 Bootstrapping process with three turker tasks



different keywords, then their similarity is given by the sum of the common keywords they share. In the example, sentences A and C would get a score of 8 (terminal) + 4 (railway station) + 3 (stop) = 15. We only use sentences that are good or acceptable representatives for this task (as judged by turkers). Alternatively, other measures of substitution overlap could be employed, e.g. as described in Erk et al. (2009). When using non-expert annotations, however, a minimum on the number of overlapping keywords is imperative to overcome the noise.

Using Chinese Whispers Biemann (2006), we apply graph clustering on this graph to group sentences that have a high similarity assuming that they contain the target word in the same meaning. This label propagation algorithm initializes every node with its own cluster label and uses a series of local update steps, in which nodes adopt the predominant label in its neighborhood.

We have chosen Chinese Whispers because it has been shown to be useful in sense clustering before (by e.g. Klapaftis and Manandhar 2008) and has the property of finding the number of clusters automatically. This is crucial since we do not know the number of senses a priori. Note, however, that we do not use the outcome of the clustering directly, but ask turkers to validate it as part of the next step in the cycle, as described in the next section. We exclude clusters consisting of singleton sentences.

For each cluster, we select as the most prototypical representative, the sentence that has the highest edge weight sum within the cluster. Ties are broken by evaluating difficulty and length (shorter is better). This sentence plus the substitutions of the cluster serves as sense inventory entry.

In case this step adds no senses to the inventory or the clustering resulted in only one sense, we continue with Task 3, otherwise, we validate the inventory.

2.2.3 *From Task 2 to Task 3*

Possible errors with clustering are either that two senses are merged or one sense is split into two entries. Overly merging results in more bootstrapping cycles, in which the 'lost' meaning not represented by the prototypical sentence is recovered. In contrast, having multiple entries per sense is more serious and must be dealt with directly. This is why we present all possible pairs of prototypical sentences in Task 2. If the majority of turkers judge that two sentences have identical meanings, we merge the entries, choosing the representative sentence at random.

Then we retrieve 100 random sentences containing our target word, group them in sets of ten and present them in Task 3.

2.2.4 *Closing the loop*

All sentences that could be matched to the inventory by the majority of turkers are added to the resource. Sentences for which the sense of the target word could not be determined due to disagreement are set aside. Sentences that are marked as uncovered by the current inventory re-enter the loop and we retrieve substitutions for them in Task 1. In our example, these might be sentences like

- D: The mid-level **station** is situated at 12,400 ft altitude.
- E: They were desperately looking for a gas **station**.

Those sentences will probably display a high overlap in substitutions with other sentences of the current or previous iterations. In this way, additional senses are identified and verified against the current inventory.

Only if almost none (we use a threshold of three) of 100 sentences are marked as uncovered, we estimate that the vast majority of the senses in the corpus are represented in our inventory, and the process terminates for the target word.

2.3 Comparison to the Semeval-07 lexical substitution task data

Since Task 1 is very similar to the annotation performed for the Semeval-07 lexical substitution task (McCarthy and Navigli 2007), we compare them in this section. We examine whether crowdsourced annotations can hold up to the quality of trained annotators as used for the Semeval task.

For the lexical substitution task annotation, five annotators (partially with linguistics background) were asked to supply from one up to three equally valid substitutions. Here, five turkers were given the task as shown in Fig. 2 with the restrictions to nouns removed. The 1710 sentences from the test section of the lexical substitution task data were used, reformatted accordingly to fit our task. As opposed to later cycles of data gathering (see next section), we used untrained workers with a minimum acceptance rate of merely 90 %.

The turker annotation can be viewed as a participating 'system' and evaluated as such, using the official scoring scripts. We examine two different scores: 'best' and 'oot'. For 'best', we only use the highest frequent turker answers (several in case of

Table 1 Scores of the turker-powered system for the Semeval-07 lexical substitution task in comparison to the best participating system in McCarthy and Navigli (2007)

System\scoring	Best	Best mode	oot	oot mode
Turker-powered	0.228	0.376	0.594	0.781
KU	0.129	0.207	0.462	0.613
UNT	0.128	0.207	0.492	0.663

ties), for 'oot' all answers were used. No duplicate answers were supplied. Table 1 shows the results that the turker-powered system would have obtained in the lexical substitution task, along with the two winning systems for comparison. See McCarthy and Navigli (2007) for details on the measures.

Not surprisingly, the turker-powered system clearly outperforms the automatic systems, especially in the 'best' evaluation. However, the scores are far from perfect. Manual inspection revealed that differences have mostly to do with sample size: especially for sentences where both professional annotators and turkers diverged much, substitutions look equally valid, but do not necessarily overlap. Some further divergence stems from the fact that the professional annotators were British English native speakers whereas our turkers are predominantly from the US.

We conclude from this that the quality of turker substitutions is reasonable. The scores reached by the turker-powered system can serve as an upper bound for automatic systems.

3 The Turk bootstrap word sense inventory

This section characterizes the TWSI 1.0. After describing the instantiation of the process that yielded the data, we describe the resource quantitatively and qualitatively. This resource has been introduced previously in Biemann and Nygaard (2010).

3.1 Experiment description

The choice of the underlying corpus was determined by our target application, a semantic search on Wikipedia. We used a sentence-segmented, POS-tagged version of English Wikipedia (dump from January 3rd, 2008) and applied a few filters to ensure complete sentences of reasonable length. Due to its diversity in topics, Wikipedia works well as a corpus for word sense acquisition. We ran the bootstrapping acquisition cycle on the 397 most frequent nouns of this corpus, implementing the crowdsourcing part with Amazon Turk. We emphasize that our methodology is not restricted to Wikipedia, in contrary to approaches like Mihalcea (2007), where internal links are utilized as sense annotations for training a WSD system. This study focuses on nouns since nouns are the most frequent part-of-speech in web queries, see Barr et al. (2008) for details.

During the annotation process of the first 50 words, the data supplied by five turkers per task was heavily curated and scammers were identified and blocked manually.

The most productive and reliable ten turkers were then invited to perform the annotation of the remaining 347 words with virtually no curation or manual intervention. With these trusted turkers, a redundancy of three assignments per task was shown to be sufficient. With the current set of trusted turkers, we were able to reach a speed of 8 words per day for the overall process. Simply adding more annotators can increase the turnaround.

The data is organized by target word. For each target word, a list of senses is provided. A sense is defined by a sense label, a prototypical sentence (the sentence used as inventory in Task 3) and a list of substitutions with their multiplicity, stemming from the data collection via Task 1. Further, sentences containing the target word (contexts) are provided with the sense labels collected in Task 3. Note that the sentences used to elicit the substitutions in Task 1 are not contained as contexts since these might have been clustered erroneously.

As an example, consider the data for the target word “magazine”, organized in two senses (194 sense-labeled sentences available):

- magazine@@1
  Sentence: Their first album was released by Columbia Records in 1972, and they were voted “Best New Band” by Creem **magazine**.
  Substitutions: publication [42], periodical [32], journal [30], manual [9], gazette [5], newsletter [4], annual [3], digest [3], circular [2]
  Number of contexts: 189
- magazine@@2
  Sentence: Instead, the film is pulled through the camera solely through the power of camera sprockets until the end, at which point springs or belts in the camera **magazine** pull the film back to the take—up side.
  Substitutions: cartridge [6], clip [5], chamber [3], holder [3], mag [3], ammunition chamber [2], cache [2], loading chamber [2]
  Number of contexts: 5

3.2 Quantitative description

Figure 6a shows the distribution of the number of cycles the words needed to converge. About two thirds of all words need only one cycle to complete, only ten words entered the cycle more than four times. The run for two words was finally terminated after ten iterations (see Sect. 3.5).

Taking a look at the granularity of the inventory, Fig. 6b shows the distribution of the number of words per number of senses. Even when using the highest frequencies nouns, almost half of the words are assigned only one sense, and over 90 % of words have fewer than five senses. For learning approaches to word sense disambiguation it is important to know how much data is available per sense. Figure 6c provides the distribution of sentences per sense in the resource. Minor senses have a low number of sentences. Targets with only one sense have almost 100 sample sentences, resulting from the 100 sentences presented per iteration in Task 3. The experiment yielded a total of 51,736 sentences with a single sense-

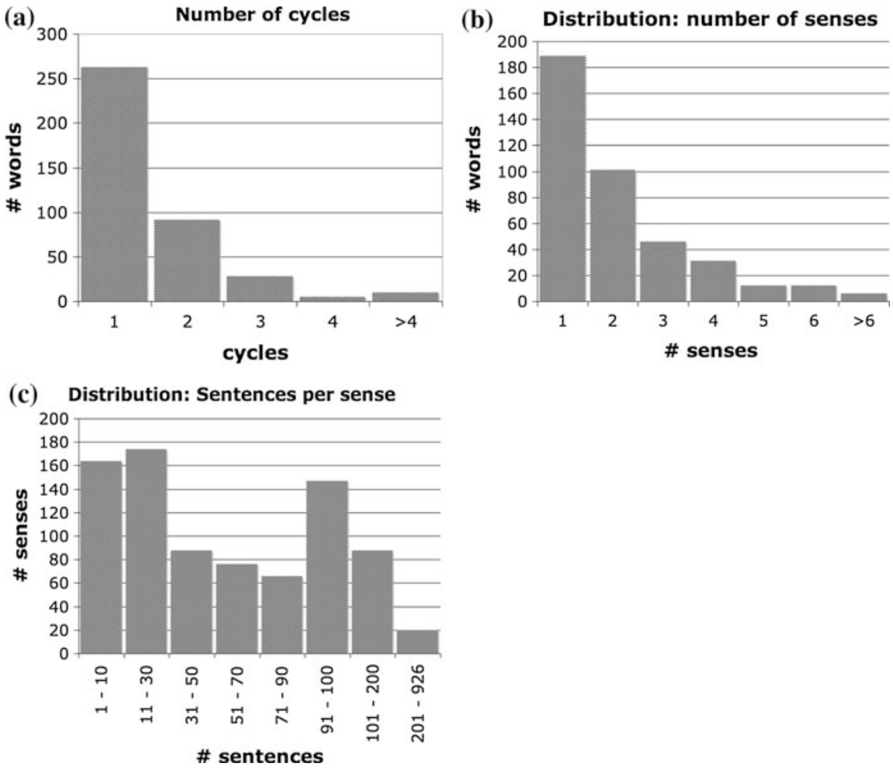
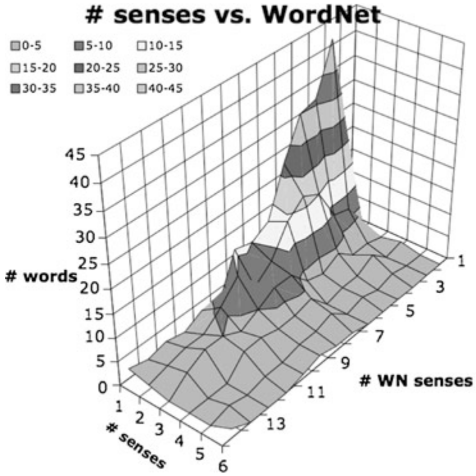


Fig. 6 Quantitative distributions. **a** Words per number of cycles until convergence. On average, a word needed 1.56 cycles to converge. **b** Number of words per number of senses. There are three words with seven senses, two words with nine senses and one word with ten senses. The average is 2.1 senses per word. **c** Number of senses per number of sentences interval. Only 5 senses have collected more than 300 sentences, at an average of 62.9 sentences per sense

labeled target word. It is therefore built into our method to create a substantial corpus that could be used for training or evaluating WSD systems, complete with a level of frequency distribution among the senses, which is valuable in its own right. We collected substitutions for a total of 8,771 sentences in Task 1. On average, a target word received 17 substitutions that were provided two or more times, and 4.5 substitutions with a frequency of ten or more. Manual inspection reveals that substitution frequencies over four are very reliable and virtually error-free.

3.3 Comparison with WordNet

Since we developed this resource in order to overcome the excessive splitting of WordNet terms into senses, we now compare the granularity of our sense inventory with WordNet. For our 397 target words, WordNet 2.1 lists 2,448 senses (excluding named entity instances), an average of 6.17 senses per target and almost three times as many senses as listed in our inventory (average number of senses: 2.1). Looking



X \ Y	1	2	3	4	5	6	≥ 7
1	20	1		1			
2	41	7	1				
3	31	8	1	3	1		
4	30	15	4	1			
5	22	17	2	1	1		
6	12	14	3				
7	10	8	7	4	3	1	
8	11	5	7	3	2	1	
9	2	10	5	3	2	2	
10	3	6	3	4	1	1	1
11	1	3	5	4	1		
12	2	3	5				1
13	1	2	3		1		1
14	2	1				2	1
≥ 15		1	3	4		5	2

Fig. 7 Number of words that have *X* WordNet senses versus *Y* senses in this inventory. 14 targets have 15 or more WordNet senses and are omitted

at the data reveals that most fine-grained WordNet distinctions have been conflated into coarser-grained senses. Also, obscure WordNet senses have not been included in our inventory. On some occasions, our inventory lists more senses, e.g. the WordNet sense *station#n#1* includes both *railway stations* and *gas stations*, whereas the crowdsourcing process distinguishes these.

Figure 7 provides a 3D-plot of the number of targets for most of the combinations of number of senses in WordNet and in our inventory. There is a direct correlation between the number of senses: words with a large number of WordNet senses also tend to be assigned a high number of senses in the crowdsourcing inventory.

3.4 Turking the turkers

We now turn to the quality of the data obtained by our acquisition cycle. Since the goal of this resource is to provide sense labels and their substitutions, we are interested in how often the substitution assigned via the sense label is acceptable.

Note that we only asked for substitutes on context for 8,771 sentences in Task 1, but project these substitutions via aggregation (clustering), inventory checking (Task 2) and matching the meaning (Task 3) to our full set of 51,736 sentences, which do not overlap with the sentences presented in Task 1. This section describes an experiment to estimate the error rate of this projection. We selected the most frequent substitution per sense and added the second and third-ranked substitution in case their frequency was three or more. This produced three substitutions for most senses and one or two substitutions for minor senses.

From our full list of sentences, we randomly sampled 500 sentences and set up an AMT task where we presented the sentence with the target word in bold along with (a) the substitutions from this projection and (b) random substitutions from the set of

Table 2 Evaluation of the substitution projection using majority vote on five turkers

	System	
	Projection	Random
Vote		
YES	469 (93.8 %)	10 (2.0 %)
NO	14 (2.8 %)	481 (96.2 %)
SOMEWHAT	17 (3.4 %)	9 (1.9 %)

Instances without majority are counted in the SOMEWHAT class

substitutions in separate tasks. Turkers were asked whether the substitutions matched the target word, matched the target word somewhat, do not match the target or the task was impossible for some reason. Table 2 shows the confusion matrix and the percentages of judgments, obtained by majority vote on five turkers. Manual checking of the positive answers for random substitutions revealed that these were in fact valid substitutions. For example, the substitutions of the municipality sense of "community" were randomly selected for a sentence containing the municipality sense of "city" as target.

Closer examination of the undecided and negative judgments for the projected substitutions showed that most of the negative judgments contained many judgments for "somewhat matching" (whereas NO answers for randomly supplied substitutions where mostly unanimous). Other sources of negative judgments included minor senses that had only substitutions with frequency one. Given that less than 3 % of projected substitutions were unanimously judged as not matching, while random substitutions were judged as not matching in over 96 % of cases, we concluded that the data produced by our process is of very high quality and is suitable for both evaluation and training Word Sense Disambiguation systems, which we will examine in Sect. 4.

3.5 Error analysis

In this section, we report findings on analyzing the bootstrapping process for a total of 100 target words. Systematic errors of the process can serve as indicators for improvements in later versions of our acquisition cycle. For the 100 targets, we observed the following problems (multitude shown in brackets):

- (4) Overlap or containment of one sense in the other leads to matching with two classes. This can be sorted out by taking into consideration the confusion matrix of meaning matching (Task 3) and the similarity of senses as measured in Task 2. An indicator of this is the number of set aside sentences in Task 3.
- (3) Systematic part-of speech tagger errors. Especially prevalent with targets that are more frequent in the non-noun reading, such as "back". Turkers did not consistently mark POS errors as impossible (although instructed). However, they reliably distinguished among senses. For example, "back in time" and "back in the yard" received separate inventory entries.
- (3) Conflation of senses. Despite differences in meaning, two senses (as perceived by us) had sufficient overlap in their substitutions to not get clustered

apart, as it happened for “relationship” in the business and personal sense. This was detected by repeatedly getting a lot of “uncovered” judgments in Task 3 yet no new senses via the substitution step in the cycle.
- (2) Oscillation of senses. Differences in the subjective judgment of turkers caused the sense inventory to oscillate between grouping and distinguishing senses, such as “over the centuries” vs. “the 16th century”. With a larger team of trusted turkers this oscillation became less of an issue since a more diverse crowd drove the process in one direction or another.

In total, we observed a successful process for about 90 % of targets, with minor problems in the remainder that seldom led to noise in the data. The following issues relate to the power-law nature of word sense distributions (cf. Kilgarriff 2004), which results in many minor senses:

- Minor senses in set aside sentences. When sampling a set of 100 sentences for Task 3, minor senses are likely to be set aside or not taken up by the clustering for lack of support. We observed this in eight targets in our analysis. While a larger sample mitigates this problem, for most applications, we are not interested in minor senses because of their low incidence and we thus do not view this as a problem. Inevitably, some minor senses in the domain did not make it into the sense inventory; however, the cases never represented more than 4 % of sample sentences.
- Few substitutions for minor senses. Of the 834 senses distinguished in our experiment, 41 did not get any substitution with frequency ≥ 2 and 142 senses did not record a substitution frequency of four or more. A way to overcome few substitutions for minor senses is to simply ask for more substitutions in the style of Task 1 for the inventory sentence or for the matched sentences for a sense in question.

From this, we conclude that a fully automatic process as set up for this study already ensures high quality. With a little manual supervision during the process, the few remaining process-related glitches can be detected early and targets can be restarted.

4 Building a system for lexical substitution

Now, we describe how we used the TWSI resource to set up a system for lexical substitution. The task we solve here has been tackled before in SemEval 2007, Task 10 (McCarthy and Navigli 2007). In this competition, all systems used predefined inventories for disambiguation and supplied substitutions associated with word senses, which is a setup we follow here: first, we disambiguate the substitution target using a WSD system, then we supply the substitutions associated with the resulting sense. The WSD system will now be described and evaluated on its own, before assessing the performance of the overall system.

Attempts to find substitutions automatically by distributional similarity over large corpora (e.g. Edmonds 1997; Inkpen 2007) did not result in sufficiently high substitution quality to use the substitutions beyond offering them to users in a thesaurus application, which is why we do not further explore this path.

4.1 Supervised WSD system

Apart from lexical features and part-of-speech (POS) sequences, successful approaches to supervised WSD employ features that aim at modeling the topicality of the context (e.g. LSA Gliozzo et al. 2005; LDA Cai et al. 2007): A context is translated into a topic vector, which informs the classifier about the general topic of the context. For example, while a monetary context for “bill” indicates its sense of “banknote”, a restaurant context would hint at its sense of “check”. These topical features are computed from a document collection and are the same for all target words. Topic Signatures (Martínez et al. 2008) is an attempt to account for differences in relevant topics per target word. Here, a large number of contexts for a given sense inventory are collected automatically using relations from a semantic resource, sense by sense. The most discriminating content words per sense are used to identify a sense in an unseen context. This approach is amongst the most successful methods in the field. It requires, however, a semantic resource of sufficient detail and size (here: WordNet) and a sense-labeled corpus to estimate priors from the sense distribution.

In the following, we describe a similar approach that uses an unlabeled corpus alone for unsupervised topic signature acquisition similar to Veronis (2004), not relying on the existence of a wordnet. Unlike previous evaluations like (Agirre et al. 2006), we do not aim to optimize the parameters for word sense induction globally, but instead offer several parameter settings to be selected by our Machine Learning setup. In this section we describe a baseline system for WSD using lexical and part-of-speech (POS) features. We further describe how to augment the baseline system with co-occurrence cluster features. The system has been described in Biemann (2010).

4.1.1 Baseline features

We created a strong baseline system with standard WSD features to compare against a system augmented with co-occurrence cluster features. Our baseline system uses 15 symbolic features per context (number of features in parentheses): (2) word forms left and right from target; (2) POS sequences left and right bigram around target; (3) POS tags of left and right word from target, and POS tag of target; (4) two left and two right nouns from target; (2) left and right verbs from target; (2) left and right adjectives from target. Note that apart from POS tagging, the baseline system does not rely on linguistic preprocessing of the data.

4.1.2 Co-occurrence cluster features

4.1.2.1 Graph preparation and parameterization Similar to the approach in Widdows and Dorow (2002), a word graph around each target word is constructed. In this work, sentence-based co-occurrence statistics from a large corpus is used as a basis to construct several word graphs for different parameterizations.

Significant co-occurrences between all content words (nouns, verbs, adjectives as identified by POS tagging) are computed from a large corpus using the tinyCC[1] tool. The full word graph for a target word is defined as all words significantly co-occurring with the target as nodes, with edge weights set to the log-likelihood significance (Dunning 1993) of the co-occurrence between the words corresponding to nodes. Edges between words that co-occur only once or with significance smaller than 6.63 (1 % confidence level) are omitted.

Aiming at different granularities of usage clusters, the graph is parameterized by a size parameter t and a density parameter n: Only the most significant t co-occurrences of the target enter the graph as nodes, and an edge between nodes is drawn only if one of the corresponding words is contained in the most significant n co-occurrences of the other.

4.1.2.2 Graph clustering parameterization As described in Biemann (2006), the neighborhood graph is clustered with Chinese Whispers. This efficient graph clustering algorithm finds the numbers of clusters automatically and returns a partition of the nodes. It is initialized by assigning different classes to all nodes in the graph. Then, a number of local update steps are performed, in which a node inherits the predominant class in its neighborhood. At this, classes of adjacent nodes are weighted by edge weight and downweighted by the degree (number of adjacent nodes) of the neighboring node. This results in hard clusters of words per target, which represent different target usages.

Downweighting nodes by degree is done according to the following intuition: nodes with high degrees are probably very universally used words and should be less influential for clustering. Three ways of node weighting are used: (a) dividing the influence of a node in the update step by the degree of the node, (b) dividing by the natural logarithm of the degree +1 and (c) not doing node weighting. The more aggressive the downweighting, the higher granularity is expected for the clustering.

Below, two different sample clusterings for the target "bank" are shown.

1. Clustering for $n = 50$, $t = 200$, weighting (a)
   - bank0: largest, north, branches, eastern, opposite, km, east, west, branch, Thames, banks, located, Danube, town, south, situated, River, Rhine, river, western, commercial, central, southern
   - bank1: right, left
   - bank2: money, robbers, deposit, robberies, cash, currency, account, deposits, Bank, robbery, funds, financial, banking, loans, notes, robber, rob, accounts, credit, assets, teller, Banco, loan, investment, savings
2. Clustering for $n = 50$, $t = 100$, weighting (c)
   - bank0: eastern, banks, central, river, km, western, south, southern, located, largest, east, deposits, commercial, Thames, north, west, Danube, town, situated, Rhine, River
   - bank1: branches, branch
   - bank2: robberies, robbers, robbery, robber

[1] http://beam.to/biem/software/TinyCC2.html.

- bank3: right, left, opposite
- bank4: loans, cash, investment, teller, account, financial, loan, deposit, credit, funds, accounts, assets, savings, banking, money, rob
- bank5: Banco, currency, notes, Bank

It can be observed that the clusterings are probably too fine-grained for word sense induction purposes, since the monetary sense can be attributed to several clusters in the second clustering. Some clusters are related to verb usages of the target, e.g. bank1 from the first clustering. Sometimes, different meanings are also lumped together in a single cluster. However, using the second clustering as a feature enables the system to assign the river bank of the Danube, given e.g. that the river bank of the Thames was found in the training.

It is emphasized that no tuning techniques are applied to arrive at the 'best' clustering. Rather, several clusterings of different granularities as *features* are made available to a supervised system. Note that this is different from Agirre et al. (2006), where a single global clustering was used *directly* in a greedy mapping to senses.

4.1.2.3 Feature assignment in context For a given occurrence of a target word, the overlap in words between the textual context and all clusters from the neighborhood graph is measured. The cluster ID of the cluster with the highest overlap is assigned as a feature. This can be viewed as a word sense induction system in its own right.

At this, several clusterings from different parameterizations are used to form distinct features, which enables the machine learning algorithm to pick the most suitable cluster features per target word when building the classification model.

4.1.2.4 Corpora for cluster features When incorporating features that are induced using large unlabeled corpora, it is important to ensure that the corpus for feature induction and the word sense labeled corpus are from the same domain, ideally from the same source.

Since TWSI has been created from Wikipedia, an English Wikipedia dump from January 2008 is used for feature induction, comprising a total of 60 million sentences. The source for the lexical sample task is the Wall Street Journal, and since the 76,400 sentences from the WSJ Penn Treebank are rather small for co-occurrence analysis, a 20 Million sentence New York Times corpus was used instead.

For each corpus, a total of 45 different clusterings were prepared for all combinations of $t = \{50,100,150,200,250\}$, $n = \{50,100,200\}$ and node degree weighting options (a), (b) and (c).

4.1.3 Machine learning setup

The classification algorithm we use in our WSD system is the AODE (Webb et al. 2005) classifier as provided by the WEKA Machine Learning software (Hall et al. 2009). This algorithm is similar to a Naïve Bayes classifier. As opposed to the latter, AODE does not assume mutual independence of features but models correlations between them explicitly, which is highly desirable in our setting since both baseline and co-occurrence cluster features are expected to be highly inter-correlated.

Further, AODE handles symbolic features, so it is directly possible for us to use lexical features and cluster IDs in the classifier. AODE showed superior performance to other classifiers handling symbolic features in preliminary experiments.

For the experiments reported below, we used tenfold cross-validation on the training for feature selection. Results are reported using held-out test data.

4.2 SemEval-07 lexical sample task

The SemEval 2007 lexical sample task (part of Task 17) provides 22,281 training sentences for 100 target words, of which 88 (35 nouns and 53 verbs) have more than one sense in the training data. The sense inventory has been provided by the OntoNotes project (Hovy et al. 2006), using high inter-annotator agreement as a guide for granularity: if fine-grained WordNet senses did not yield high enough agreement with human annotators, they were grouped together. The average number of senses per ambiguous word in the training is 3.95, compared to the 3.91 average senses per word in the TWSI.

We ran tenfold cross validation on the training sentences for ambiguous words, adding all 45 co-occurrence cluster features one at the time. All systems with single cluster features outperformed the baseline system precision of 87.1 %, ranging from 88.0 to 88.3 % precision. For combining the best k single cluster features for $k = \{2,3,5,10\}$, the best performing system resulted in a tenfold precision of 88.5 % for $k = 3$, showing significant gain over the baseline. This system is used for evaluation on the official test set.

We used the system configuration determined from the training data, trained it on the full training set and applied it to the test data provided by the task organizers.

Since our AODE classifier reports a confidence score (corresponding to the class probability for the winning class at classification time), we are able to investigate a tradeoff between Precision P and Recall R to optimize the F1-value[2] used for scoring in the lexical sample task. Table 3 shows the results for the baseline and the system augmented with the top 3 cluster features in comparison with the two best systems in the 2007 evaluation, both for maximal recall and for the optimal F1-value on the test data of 4,851 labeled contexts, which was merely used for evaluation to create the same conditions that held for the participating systems of this task.

It is surprising that the baseline system outperforms the second-best system in the 2007 evaluation. This might be attributed to the AODE classifier used, but also hints at the power of symbolic lexical features in general. The co-occurrence cluster system outperforms the baseline, but does not reach the performance of the winning NUS-ML system. This system uses a variety of standard features plus (global) topic features via LDA and features from dependency parses. However, all reported systems fall into each other's error margins, unlike when evaluating on training data splits. We conclude from this that our WSD setup is competitive to other WSD

[2] $F1 = \frac{2PR}{P+R}$.

Table 3 Cluster co-occurrence features and baseline in comparison to the best two systems in the SemEval 2007 Task 17 lexical sample evaluation (Pradhan et al. 2007): NUS-ML (Cai et al. 2007) and UBC-ALM (Agirre and Lopez de Lacalle 2007)

System	NUS-ML	Top3 cluster optimal F1	Top3 cluster max recall	Baseline optimal F1	Baseline max recall	UBC-ALM
F1 value	88.7 % ± 1.2	88.0 % ± 1.2	87.8 % ± 1.2	87.5 % ± 1.2	87.3 % ± 1.2	86.9 % ± 1.2

Error margins provided by the task organizers

systems in the literature, while using only minimal linguistic preprocessing and no word sense inventory information beyond what is provided by training examples.

4.3 Lexical substitution system

Having substitutions associated with word senses in the TWSI, we can set up a lexical substitution system by first disambiguating a target word to assign a TWSI sense, then supplying the associated highest ranked substitutions in context. We will report evaluation results for both steps separately.

4.3.1 Word sense disambiguation on the TWSI

Figure 8 shows the learning curve for various amounts of randomly selected training/test splits, averaged over three runs each with different random seeds. The most frequent sense (MFS) baseline hovers around 71.5 %, the baseline system gets 80.7 % precision at 90 % training. For both corpora, the top performing single cluster feature significantly improves over the baseline: The best single cluster feature achieves 82.9 % precision. Increasing the number of cluster features leads first to improvements, then to a slight degradation for k = 10.

The best system using co-occurrence cluster features is obtained by adding the top 5 single co-occurrence cluster features to the baseline features with a tenfold cross validation score of 83.0 %. This system is used in the substitution evaluation experiment described below. While the difference in the 90 % training situation is very small, the systems using several co-occurrence cluster features excel in reduced-training situations. Looking at the parameterizations of the features, we found no general trend for either parameter n, t or node weighting. Overall, the cluster features reduce the amount of training data needed to reach equal levels of performance to less than half, see Fig. 8.

4.3.2 Substitution acceptability evaluation

We now provide evaluation results for substitution acceptability of our overall lexical substitution system on the TWSI. We randomly selected 500 labeled sentences from the overall data (for all 397 nouns, not just the ambiguous nouns used in the experiments above) and used the tenfold test classifications as described in the previous section for word sense assignment.

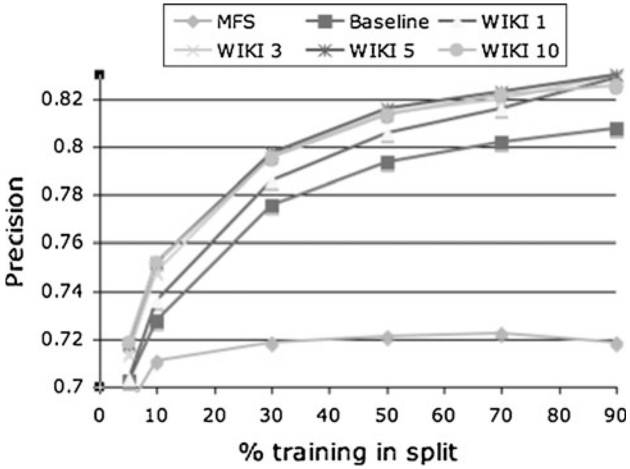


Fig. 8 Learning curve on WSD quality for different systems for the TWSI data

Then, we used the same crowdsourcing task as in Sect. 3.4 for measuring the system's substitution acceptability.

Table 4 shows the results for averaging over the worker's responses. For being counted as belonging to the YES or NO class, the majority of workers had to choose this option; otherwise the item was counted into the SOMEWHAT class. The substitution quality of the gold standard is somewhat noisy, containing 2.8 % errors and 3.4 % questionable cases. Despite this, our system is able to assign acceptable substitutions in over 91 % of cases, questionable substitutions for 3.4 % at an error rate of only 5.4 %. Checking the positively judged random assignments, an acceptable substitution was found in about half of the cases by manual inspection, which allows us to estimate the worker noise at about 1 %. Note that the fraction of acceptable substitutions is only slightly lower than the substitution projection of the TWSI as measured in Sect. 3.4.

Now we investigate the precision-recall tradeoff in reduced-recall settings. In applications, it might make sense to only return substitutions when the system is relatively sure. We used the confidence values of our AODE classifier to control recall and provide precision scores for different coverage (percentage of attempted classifications) levels in Table 5.

Table 4 Substitution acceptability as measured by crowdsourcing for system assignments and random assignments

	System	
	Classifier	Random
Vote		
YES	456 (91.2 %)	12 (2.4 %)
NO	27 (5.4 %)	485 (97.0 %)
SOMEWHAT	17 (3.4 %)	3 (0.6 %)

Table 5 Substitution acceptability in reduced coverage settings

Coverage	100 (%)	95 (%)	90 (%)	80 (%)	70 (%)
YES	91.2	91.8	93.8	94.8	95.7
NO	5.4	3.4	2.9	2.0	0.9

SOMEWHAT class accounts for percentage points missing to 100 %

Dependent on the application needs, the confidence score of the classifier allows the system to reduce the error rate of the supplied substitutions. For example at a coverage level of 70 %, not even 1 % of system substitutions were judged as incorrect.

5 Conclusion

In this article, we have demonstrated how to create a high quality semantic resource from scratch. Using Amazon Turk as a crowdsourcing platform and breaking down the problem of substitution and word sense inventory acquisition into three simple tasks, we were able to produce a rich semantic resource for semantic indexing at a comparatively low cost. Further contributions of this work are the definition of word senses according to substitutability equivalence and the usage of a bootstrapping process depending on human input.

Compared to WordNet, which is the most commonly used inventory for word sense disambiguation, our resource has a much richer set of sample usages, a larger set of substitutions, fewer fine-grained distinctions and provides a corpus-based estimate on word sense distribution. Our method does not need pre-processing other than lemmatizing and POS tagging and can be directly applied to other domains or languages. We have run a pilot study for verb sense acquisition with equally encouraging results.

Empowered with this resource, we have trained a system for lexical substitutions that proceeds in two steps: First, the sense of a target word is disambiguated, then the respective substitutions are provided.

For supervised WSD, we set up a strong baseline system and evaluated the influence of co-occurrence cluster features, which are obtained by clustering the co-occurrence neighborhood graphs of target words obtained from a large, unlabeled corpus. Evaluating against a standard dataset, we were able to show that our approach to WSD is competitive and that cluster co-occurrence features yield improvements over the baseline. We measured performance of our lexical substitution system by having the acceptability of the system-provided substitutions in context manually judged. With error rates in the single figures and the possibility to reduce errors further by sacrificing recall, we provide a firm enabling technology for semantic search or other applications.

To test the scalability of the approach, another 617 nouns were added as TWSI 2.0 in order of decreasing frequency. A workforce of about 25 active annotators completed about 8 words per day on average. This already demonstrates the scalability; adding more annotators can even further increase creation speed.

Further work can proceed along two lines: On the one hand, one can explore how to enrich the resource itself, e.g. by expanding the target set, acquisition of hypernyms or other relations in Task 1 style, for example, or by creating synsets of senses with the same substitutions that can substitute for each other. We feel, however, that this always should be driven by a specific task.

The data described in this paper is available for download[3] under a Creative Commons License. A software package that realizes the lexical substitution system described in this paper is available for download[4] under the Gnu Public License.

References

Agirre, E., & Edmonds, P. (Eds.). (2006). *Word sense disambiguation: Algorithms and applications, volume 33 of text, speech and language technology*. New York: Springer.

Agirre, E., & Lopez de Lacalle, O. (2007). UBC-ALM: Combining k-NN with SVD for WSD. In *Proceedings of the fourth international workshop on semantic evaluations* (*SemEval-2007*) (pp. 342–345). Prague, Czech Republic.

Agirre, E., Martínez, D., Lopez de Lacalle, O., & Soroa, A. (2006). Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In *Proceedings of TextGraphs: The second workshop on graph based methods for natural language processing* (pp. 89–96). New York City, USA.

Barr, C., Jones, R., & Regelson, M. (2008). The linguistic structure of English web-search queries. In *Proceedings of EMNLP 2008* (pp. 1021–1030). Honolulu, HI, USA.

Biemann, C. (2006). Chinese whispers—an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the HLT-NAACL-06 workshop on Textgraphs-06*. New York, USA.

Biemann C. (2010). Co-occurrence cluster features for lexical substitutions in context. In *Proceedings of the ACL-2010 Workshop on Textgraphs*. Uppsala, Sweden.

Biemann, C., & Nygaard, V. (2010). Crowdsourcing WordNet. In *Proceedings of the 5th global WordNet conference, Mumbai, India*. ACL Data and Code Repository, ADCR2010T005.

Cai, J. F., Lee, W. S., & Teh, Y. W. (2007). NUS-Ml: Improving word sense disambiguation using topic features. In *Proceedings of the fourth international workshop on semantic evaluations* (*SemEval-2007*) (pp. 249–252). Prague, Czech Republic.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics, 19*(1), 61–74.

Edmonds, P. (1997). Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the 35th annual meeting of the association for computational linguistics* (pp. 507–509). Madrid, Spain.

Erk, K., McCarthy, D., & Gaylord, N. (2009). Investigations on word senses and word usages. In *Proceedings of the joint conference of the 47th annual meeting of the association for computational linguistics and the 4th international joint conference on natural language processing of the asian federation of natural language processing ACL-IJCNLP, Singapore.*

Gliozzo, A., Giuliano, C., & Strapparava, C. (2005). Domain kernels for word sense disambiguation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 403–410). Morristown, NJ, USA.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1), 10–18.

Hovy, E., Marcus, M., Palmer, M. Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% solution. In *Proceedings of HLT-NAACL 2006* (pp. 57–60).

[3] http://www.ukp.tu-darmstadt.de/data/twsi-lexical-substitutions/.

[4] http://www.ukp.tu-darmstadt.de/software/twsi-sense-substituter/.

Inkpen, D. (2007). Near-synonym choice in an intelligent thesaurus. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics* (*NAACL*)*; Proceedings of the Main Conference* (pp. 356–363). Rochester, NY, USA.

Kilgarriff, A. (2004). How dominant is the commonest sense of a word. In *Proceedings of text, speech, dialogue* (pp. 1–9). Springer-Verlag.

Klapaftis, I. P., & Manandhar, S. (2008). Word sense induction using graphs of collocations. In *Proceedings of the 18th European conference on artificial intelligence* (*ECAI-2008*). Patras, Greece: IOS Press.

Martínez, D., Lopez de Lacalle, O., & Agirre, E. (2008). On the use of automatically acquired examples for all-nouns word sense disambiguation. *Journal of Artificial Intelligence (JAIR), 33*, 79–107.

McCarthy, D., & Navigli, R. (2007). Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the fourth international workshop on semantic evaluations* (SemEval-2007) (pp. 48–53), Prague, Czech Republic.

Mihalcea, R. (1998). SEMCOR semantically tagged corpus. Unpublished manuscript.

Mihalcea, R. (2007). Using Wikipedia for automatic word sense disambiguation. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics* (*NAACL*)*, Rochester, NY, USA*.

Mihalcea, R., & Moldovan, D. (2001). Automatic generation of a coarse grained WordNet. In *Proceedings of the NAACL worshop on WordNet and other lexical resources, Pittsburg, USA*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography, 3*, 235–244.

Pradhan, S., Loper, E., Dligach, D., & Palmer, M. (2007). SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proceedings of the fourth international workshop on semantic evaluations* (*SemEval-2007*) (pp. 87–92). Prague, Czech Republic.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of the 17th annual international ACM-SIGIR conference on research and development in information retrieval*. Dublin, Ireland, 3–6 July 1994 (Special Issue of the SIGIR Forum), pp. 142–151. New York: ACM/Springer.

Sanderson, M. (2000). Retrieving with good sense. *Information Retrieval, 2*(1), 49–69.

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *Conference on empirical methods in natural language processing, EMNLP 2008, proceedings of the conference* (pp. 254–263), 25–27 October 2008, Honolulu, Hawaii, USA.

Stokoe, C. (2005). Differentiating homonymy and polysemy in information retrieval. In *Proceedings of EMNLP-HLT*. Vancouver, BC, Canada.

Veronis, J. (2004). Hyperlex: Lexical cartography for information retrieval. *Computer Speech & Language, 18*(3), 223–252.

Webb, G., Boughton, J., & Wang, Z. (2005). Not so Naive Bayes: Aggregating one-dependence estimators. *Machine Learning, 58*(1), 5–24.

Widdows, D., & Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on computational linguistics* (pp. 1–7). Morristown, NJ, USA.