Evaluation Metrics for the Lexical Substitution Task

Sanaz Jabbari  Mark Hepple  Louise Guthrie
Department of Computer Science, University of Sheffield
211 Portobello Street, Sheffield, S1 4DP, UK
{S.Jabbari,M.Hepple,L.Guthrie}@dcs.shef.ac.uk

Abstract
We identify some problems of the evaluation metrics used for the English Lexical Substitution Task of SemEval-2007, and propose alternative metrics that avoid these problems, which we hope will better guide the future development of lexical substitution systems.

1 Introduction
The English Lexical Substitution task at SemEval-2007 (here called ELS07) requires systems to find substitutes for target words in a given sentence (McCarthy & Navigli, 2007: M&N). For example, we might replace the target word match with game in the sentence they lost the match. System outputs are evaluated against a set of candidate substitutes proposed by human subjects for test items. Targets are typically sense ambiguous (e.g. match in the above example), and so task performance requires a combination of word sense disambiguation (by exploiting the given sentential context) and (near) synonym generation. In this paper, we discuss some problems of the evaluation metrics used in ELS07, and then propose some alternative measures that avoid these problems, and which we believe will better serve to guide the development of lexical substitution systems.

2 Evaluation Materials
Briefly stated, the ELS07 dataset comprises around 2000 sentences, providing 10 test sentences each for some 201 preselected target words, which were required to be sense ambiguous and have at least one synonym, and which include nouns, verbs, adjectives and adverbs. Five human annotators were asked to suggest up to three substitutes for the target word of each test sentence, and their collected suggestions serve as the gold standard against which system outputs are compared. Around 300 sentences were distributed as development data, and the remainder retained for the final evaluation.

To assist defining our metrics, we formally describe this data as follows.\(^2\) For each sentence \(t_i\) in the test data (\(1 \leq i \leq N\), \(N\) the number of test items), let \(H_i\) denote the set of human proposed substitutes. A key aspect of the data is the count of human annotators that proposed each candidate (since a term appears a stronger candidate if proposed by annotators). For each \(t_i\), there is a function \(freq_i\) which returns this count for each term within \(H_i\) (and 0 for any other term), and a value \(maxfreq_i\) corresponding to the maximal count for any term in \(H_i\). The pairing of \(H_i\) and \(freq_i\) in effect provides a multiset representation of the human answer set. We
dcan produce. In what follows, we address these two cases in separate sections, and then present some results for applying our new metrics for the second case. We begin by briefly introducing the test materials that were created for the ELS07 evaluation.
use \(|S|^i\) in what follows to denote the multiset cardinality of \(S\) according to \(freq_i\), i.e. \(\sum_{a \in S} freq_i(a)\). Some of the ELS07 metrics use a notion of mode answer \(m_i\), which exists only for test items that have a single most-frequent human response, i.e. a unique \(a \in H_i\) such that \(freq_i(a) = \maxfreq_i\). To adapt an example from M&N, an item with target word happy (adj) might have human answers \{'glad, merry, sunny, jovial, cheerful\}' with counts \((3,3,2,1,1)\) respectively. We will abbreviate this answer set as \(H_i = \{G:3,M:3,S:2,J:1,Ch:1\}\) where it is used later in the paper.

3 Best Answer Measures

Two of the ELS07 tasks address how well systems are able to find a ‘best’ substitute for a test item, for which individual test items are scored as follows:

\[
best(i) = \frac{\sum_{a \in A_i} freq_i(a)}{|H_i|^i \times |A_i|}
\]

\[
mode(i) = \begin{cases} 
1 & \text{if } \maxfreq_i = m_i \\
0 & \text{otherwise}
\end{cases}
\]

For the first task, a system can return a set of answers \(A_i\) (the answer set for item \(i\)), but since the score achieved is divided by \(|A_i|\), returning multiple answers only serves to allow a system to ‘hedge its bets’ if it is uncertain which candidate is really the best. The optimal score on a test item is achieved by returning a single answer whose count is \(\maxfreq_i\), with proportionately lesser credit being received for any answer in \(H_i\) with a lesser count. For the second task, which uses the mode metric, only a single system answer – its ‘best guess’ \(bg_i\) – is allowed, and the score is simply 0 or 1 depending on whether the best guess is the mode. Overall performance is computed by averaging across a broader set of test items (which for the second task includes only items having a mode value). M&N distinguish two overall performance measures: Recall, which averages over all relevant items, and Precision, which averages only over those items for which the system gave a non-empty response.

We next discuss these measures and make an alternative proposal. The task for the first measure seems a reasonable one, i.e. assessing the ability of systems to provide a ‘best’ answer for a test item, but allowing them to offer multiple candidates (to ‘hedge their bets’). However, the metric is unsatisfactory in that a system that performs optimally in terms of this task (i.e. which, for every test item, returns a single correct ‘most frequent’ response) will get a score that is well below 1, because the score is also divided by \(|H_i|^i\), the multiset cardinality of \(H_i\), whose size varies between test items (being a consequence of the number of alternatives suggested by the human annotators), but which is typically larger than the numerator value \(\maxfreq_i\) of an optimal answer (unless \(H_i\) is singleton). This problem is fixed in the following modified metric definition, by dividing instead by \(\maxfreq_i\), as then a response containing a single optimal answer will score 1.

\[
\text{best}(i) = \frac{\sum_{a \in A_i} freq_i(a)}{\maxfreq_i \times |A_i|} \quad \text{best}_1(i) = \frac{freq(bg_i)}{\maxfreq_i}
\]

With \(H_i = \{G:3,M:3,S:2,J:1,Ch:1\}\), for example, an optimal response \(A_i = \{M\}\) receives score 1, where the original metric gives score 0.3. Singleton responses receiving a correct but non-optimal answer receive proportionately lower credit, e.g. for \(A_i = \{S\}\) we score 0.66 (vs. 0.2 for the original metric). For a non-singleton answer set including, say, a correct answer and an incorrect one, the credit for the correct answer will be halved, e.g. for \(A_i = \{S, X\}\) we score 0.33.

Regarding the second task, we think it reasonable to have a task where systems may offer only a single ‘best guess’ response, but argue that the mode metric used has two key failings: it is too brittle in being applicable only to items that have a mode answer, and it loses information valuable to system ranking, in assigning no credit to a response that might be good but not optimal. We propose instead the \(\text{best}_1\) metric above, which assigns score 1 to a best guess answer with count \(\maxfreq_i\), but applies to all test items irrespective of whether or not they have a unique mode. For answers having lesser counts, proportionately less credit is assigned. This metric is equivalent to the new \(\text{best}\) metric shown beside it for the case where \(|A_i| = 1\).

For assessing overall performance, we suggest just taking the average of scores across all test items, c.f. M&N’s Recall measure. Their Precision metric is presumably intended to favour a system that can tell whether it does or does not have any good answers to return. However, the ability to draw a
boundary between good vs. poor candidates will be reflected widely in a system's performance and captured elsewhere (not least by the coverage metrics discussed later) and so, we argue, does not need to be separately assessed in this way. Furthermore, the fact that a system does not return any answers may have other causes, e.g. that its lexical resources have failed to yield any substitution candidates for a term.

4 Measures of Coverage

A third task of ELS07 assesses the ability of systems to field a wider set of good substitution candidates for a target, rather than just a ‘best’ candidate. This ‘out of ten’ (oot) task allows systems to offer a set \( A_i \) of upto 10 guesses per item \( i \), and is scored as:

\[
\text{oot}(i) = \frac{\sum_{a \in A_i} \text{freq}(a)}{|H_i|^i}
\]

Since the score is not divided by the answer set size \( |A_i| \), no benefit derives from offering less than 10 candidates.\(^3\) When systems are asked to field a broader set of candidates, we suggest that evaluation should assess if the response set is good in containing as many correct answers as possible, whilst containing as few incorrect answers as possible. In general, systems will tackle this problem by combining a means of ranking candidates (drawn from lexical resources) with a means of drawing a boundary between good and bad candidates, e.g. threshold setting.\(^4\) Since the oot metric does not penalise incorrect answers, it does not encourage systems to develop such boundary methods, even though this is important to their ultimate practical utility.

The view of a ‘good’ answer set described above suggests a comparison of \( A_i \) to \( H_i \) using versions of ‘recall’ and ‘precision’ metrics, that incorporate the ‘weighting’ of human answers via \( \text{freq}_i \). Let us begin by noting the obvious definitions for recall and precision metrics without count-weighting:

\[
R(i) = \frac{|H_i \cap A_i|}{|H_i|^i} \quad P(i) = \frac{|H_i \cap A_i|}{|A_i|^i}
\]

Our definitions of these metrics, given below, do include count-weighting, and require some explanation. The numerator of our recall definition is \( |A_i|^i \) not \( |H_i \cap A_i|^i \) as \( |A_i|^i = |H_i \cap A_i|^i \) (as \( \text{freq}_i \) assigns 0 to any term not in \( H_i \)), an observation which also affects the numerator of our \( P \) definition. Regarding the latter’s denominator, merely dividing by \( |A_i|^i \) would not penalise incorrect terms (as, again, \( \text{freq}_i(a) = 0 \) for any \( a \notin H_i \)), so this is done directly by adding \( k|A_i - H_i| \), where \( |A_i - H_i| \) is the number of incorrect answers, and \( k \) some penalty factor, which might be \( k = 1 \) in the simplest case. (Note that our weighted \( R \) metric is in fact equivalent to the oot definition above.) As usual, an \( F \)-score can be computed as the harmonic mean of these values (i.e. \( F = 2PR/(P + R) \)). For assessing overall performance, we might average \( P \), \( R \) and \( F \) scores across all test items.

\[
R(i) = \frac{|A_i|^i}{|H_i|^i} \quad P(i) = \frac{|A_i|^i}{|A_i|^i + k|A_i - H_i|}
\]

With \( H_i = \{G:3,M:3,S:2,J:1,Ch:1\} \), for example, the perfect response set \( A_i = \{G,M,S,J,Ch\} \) gives \( P \) and \( R \) scores of 1. The response \( A_i = \{G,M,S,J,Ch,X,Y,Z,V,W\} \), containing all correct answers plus 5 incorrect ones, gets \( R = 1 \), but only \( P = 0.66 \) (assuming \( k = 1 \), giving \( 10/(10 + 5) \)). The response \( A_i = \{G,S,J,X,Y\} \), with 3 out of 5 correct answers, plus 2 incorrect ones, gets \( R = 0.6 \) \((6/10)\) and \( P = 0.75 \) \((6/6 + 2)\)

5 Applying the Coverage measure

Although the ‘best guess’ task is a valuable indicator of the likely utility of a lexical substitution system within various broader applications, we would argue that the core task for lexical substitution is coverage, i.e. the ability to field a broad set of correct substitution candidates. This task requires systems both to field and rank promising candidates, and to have a means of drawing a boundary between the good and bad candidates, i.e. a boundary strategy.

In this section, we apply the coverage metrics to the outputs of some lexical substitution systems, and
Table 1: Out-of-ten recall scores for all the systems (with a subdivision by pos of target item).

| Model | All words | By part-of-speech |
|-------|-----------|-------------------|
|       | nouns | adj | verb | adv |
| bow   | .326  | .343 | .334 | .205 | .461 |
| lm    | .393  | .372 | .422 | .252 | .562 |
| cmlc  | .414  | .404 | .447 | .311 | .534 |
| KU    | .462  | .408 | .511 | .398 | .567 |

Table 2: Optimal F-scores (macro-avgd) for coverage, computed over the (oot) ranked outputs of the systems (with penalty factor \( k = 1 \)).

| Model | All words | By part-of-speech |
|-------|-----------|-------------------|
|       | nouns | adj | verb | adv |
| bow   | .298  | .315 | .302 | .189 | .422 |
| lm    | .371  | .35  | .408 | .24  | .539 |
| cmlc  | .395  | .383 | .419 | .31  | .506 |
| KU    | .435  | .379 | .477 | .385 | .536 |

Table 3: Coverage F-scores (macro-avgd), for simple boundary strategies (with penalty factor \( k = 1 \)).

| Model | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|-------|----|----|----|----|----|----|----|----|----|----|
| bow   | .067| .114| .151| .173| .191| .201| .212| .219| .222| .225|
| lm    | .119| .192| .228| .246| .256| .267| .271| .272| .271| .271|
| cmlc  | .139| .205| .251| .271| .284| .288| .291| .290| .289| .286|
| KU    | .173| .244| .287| .307| .318| .321| .320| .318| .314| .311|

Our first problem is that these systems are developed for the oot task, not coverage, so after ranking their candidates, they do not attempt to draw a boundary between the candidates worth returning and those not. Instead, we here use the oot outputs to compute an optimal performance for each system, i.e. we find, for the ranked candidates of each question, the cut-off position giving the highest F-score, and then average these scores across questions, which tells us the F-score the system could achieve if it had an optimal boundary strategy. These scores, shown in Table 2, indicate a ranking of systems in line with that in Table 1, which is not surprising as both will ultimately reflect the quality of candidate ranking achieved by the systems.

Table 3 shows the coverage results achieved by applying a naive boundary strategy to the system outputs. The strategy is just to always return the top \( n \) candidates for each question, for a fixed value \( n \). Again, performance correlates straightforwardly with the underlying quality of ranking. Comparing tables, we see, for example, that by always returning 6 candidates, the system KU could achieve a coverage of .32 as compared to the .435 optimal score.

References

D. McCarthy and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. Proc. of the 4th Int. Workshop on Semantic Evaluations (SemEval-2007), Prague.

S. Jabbari. 2010. A Statistical Model of Lexical Context, PhD Thesis, University of Sheffield.

S. Jabbari, M. Hepple and L. Guthrie. 2010. Evaluating Lexical Substitution: Analysis and New Measures. Proc. of the 7th Int. Conf. on Language Resources and Evaluation (LREC-2010). Malta.

D. Yuret. 2007. KU: Word Sense Disambiguation by Substitution. In Proc. of the 4th Int. Workshop on Semantic Evaluations (SemEval-2007), Prague.