Named Entity Scoring for Speech Input

John D. Burger, David Palmer, Lynette Hirschman
The MITRE Corporation, 202 Burlington Road, Bedford, MA 01730, USA
john@mitre.org, palmer@mitre.org, lynette@mitre.org

Abstract

This paper describes a new scoring algorithm that supports comparison of linguistically annotated data from noisy sources. The new algorithm generalizes the Message Understanding Conference (MUC) Named Entity scoring algorithm, using a comparison based on explicit alignment of the underlying texts, followed by a scoring phase. The scoring procedure maps corresponding tagged regions and compares these according to tag type and tag extent, allowing us to reproduce the MUC Named Entity scoring for identical underlying texts. In addition, the new algorithm scores for *content* (transcription correctness) of the tagged region, a useful distinction when dealing with noisy data that may differ from a reference transcription (e.g., speech recognizer output). To illustrate the algorithm, we have prepared a small test data set consisting of a careful transcription of speech data and manual insertion of SGML named entity annotation. We report results for this small test corpus on a variety of experiments involving automatic speech recognition and named entity tagging.

1. Introduction: The Problem

Linguistically annotated training and test corpora are playing an increasingly prominent role in natural language processing research. The Penn TREEBANK and the SUSANNE corpora (Marcus 93, Sampson 95) have provided corpora for part-of-speech taggers and syntactic processing. The Message Understanding Conferences (MUCs) and the Tipster program have provided corpora for newswire data annotated with named entities[1] in multiple languages (Merchant 96), as well as for higher level relations extracted from text. The value of these corpora depends critically on the ability to evaluate hypothesized annotations against a gold standard reference or key.

To date, scoring algorithms such as the MUC Named Entity scorer (Chinchor 95) have assumed that the documents to be compared differ only in linguistic annotation, not in the underlying text.[2] This has precluded applicability to data derived from noisy sources. For example, if we want to compare named entity (NE) processing for a broadcast news source, created via automatic speech recognition and NE tagging, we need to compare it to data created by careful human transcription and manual NE tagging. But the underlying texts—the recognizer output and the gold standard transcription—differ, and the MUC algorithm cannot be used. Example 1 shows the reference transcription from a broadcast news source, and below it, the transcription produced by an automatic speech recognition system. The excerpt also includes reference and hypothesis NE annotation, in the form of SGML tags, where <P> tags indicate the name of a person, <L> that of a location, and <O> an organization.[3]

We have developed a new scoring algorithm that supports comparison of linguistically annotated data from noisy sources. The new algorithm generalizes the MUC algorithm, using a comparison based on explicit alignment of the underlying texts. The scoring procedure then maps corresponding tagged regions and compares these according to tag type and tag extent. These correspond to the components currently used by the MUC scoring algorithm. In addition, the new algorithm also compares the content of the tagged region, measuring correctness of the transcription within the region, when working with noisy data (e.g., recognizer output).

2. Scoring Procedure

The scoring algorithm proceeds in five stages:

1. Preprocessing to prepare data for alignment
2. Alignment of lexemes in the reference and hypothesis files
3. Named entity mapping to determine corresponding phrases in the reference and hypothesis files
4. Comparison of the mapped entities in terms of tag type, tag extent and tag content
5. Final computation of the score

[1]MUC "named entities" include person, organization and location names, as well as numeric expressions.

[2]Indeed, the Tipster scoring and annotation algorithms require, as part of the Tipster architecture, that the annotation preserve the underlying text including white space. The MUC named entity scoring algorithm uses character offsets to compare the mark-up of two texts.

[3]The SGML used in Tipster evaluations is actually more explicit than that used in this paper, e.g., <ENAMEX TYPE=PERSON> rather than <P>.

```
ref: AT THE <L> NEW YORK </L> DESK I'M <P> PHILIP BOROFF </P> <L> MISSISSIPPI </L> REPUBLICAN
hyp: AT THE <L> NEWARK </L> BASK ON FILM FORUM MISSES THE REPUBLICAN
```

Example 1: Aligned and tagged text

2.1 Stage 1: Preprocessing

The algorithm takes three files as input: the human-transcribed reference file with key NE phrases, the speech recognizer output, which includes coarse-grained timestamps used in the alignment process, and the recogizer output tagged with NE mark-up.

The first phase of the scoring algorithm involves reformatting these input files to allow direct comparison of the raw text. This is necessary because the transcript file and the output of the speech recognizer may contain information in addition to the lexemes. For example, for the Broadcast News corpus provided by the Linguistic Data Consortium,[4] the transcript file contains, in addition to mixed-case text representing the words spoken, extensive SGML and pseudo-SGML annotation including segment timestamps, speaker identification, background noise and music conditions, and comments. In the preprocessing phase, this annotation and all punctuation is removed, and all remaining text is converted to upper-case. Each word in the reference text is then assigned an estimated timestamp based on the explicit timestamp of the larger parent segment.[5]

Given the sequence of all the timestamped words in each file, a coarse segmentation and alignment is performed to assist the lexeme alignment in Stage 2. This is done by identifying sequences of three or more identical words in the reference and hypothesis transcriptions, transforming the long sequence into a set of shorter sequences, each with possible mismatches. Lexeme alignment is then performed on these short sequences.[6]

2.2 Stage 2: Lexeme Alignment

A key component of the scoring process is the actual alignment of individual lexemes in the reference and hypothesis documents. This task is similar to the alignment that is used to evaluate word error rates of speech recognizers: we match lexemes in the hypothesis text with their corresponding lexemes in the reference text. The standard alignment algorithm used for word error evaluation is a component of the NIST SCLite scoring package used in the Broadcast News evaluations (Garofolo 97). For each lexeme, it provides four possible classifications of the alignment: correct, substitution, insertion, and deletion. This classification has been successful for evaluating word error. However, it restricts alignment to a one-to-one mapping between hypothesis and reference texts. It is very common for multiple lexemes in one text to correspond to a single lexeme in the other, in addition to multiple-to-multiple correspondences. For example, compare `New York` and `Newark` in Example 1. Capturing these alignment possibilities is especially important in evaluating NE performance, since the alignment facilitates phrase mapping and comparison of tagged regions.

```
ref: AT THE NEW YORK DESK I'M PHILIP BOROFF MISSISSIPPI REPUBLICAN
hyp: AT THE NEWARK BASK ON FILM FORUM MISSES THE REPUBLICAN

ref: AT THE NEW YORK DESK I'M PHILIP BOROFF MISSISSIPPI REPUBLICAN
hyp: AT THE NEWARK BASK ON FILM FORUM MISSES THE REPUBLICAN
```

Example 2: SCLite alignment (top) vs. phonetic alignment (bottom)

In the current implementation of our scoring algorithm, the alignment is done using a phonetic alignment algorithm (Fisher 93). In direct comparison with the standard alignment algorithm in the SCLite package, we have found that the phonetic algorithm results in more intuitive results. This can be seen clearly in Example 2, which repeats the reference and hypothesis texts of the previous example. The top alignment is that produced by the SCLite algorithm; the bottom by the phonetic algorithm. Since this example contains several instances of potential named entities, it also illustrates the impact of different alignment algorithms (and alignment errors) on phrase mapping and comparison. We will compare the effect of the two algorithms on the NE score in Section 3.

[4] http://www.ldc.upenn.edu/

[5] It should be possible to provide more accurate word timestamps by using a large-vocabulary recognizer to provide a forced alignment on the clean transcription.

[6] The sequence length is dependent on the word-eror rate of the recognizer ouput, but in general the average sequence is 20–30 words long after this coarse segmentation.

```
ref:  INVESTING  *        *         *      AND     TRADING  WITH  CUBA   FROM  OTTAWA  THIS  IS
hyp:  INVESTING  IN   TRAINING  WOULD   KEEP      OFF      A     LOT    OF    WHAT    THIS  IS

ref:  INVESTING  AND  TRADING  WITH  CUBA     *     FROM  OTTAWA  THIS  IS
hyp:  INVESTING  IN   TRAINING WOULD KEEP OFF  A     LOT   OF WHAT THIS  IS
```

Example 3: Imperfect alignments (SCLite top, phonetic bottom)

Even the phonetic algorithm makes alignment mistakes. This can be seen in Example 3, where, as before, SCLite's alignment is shown above that of the phonetic algorithm. Once again, we judge the latter to be a more intuituive alignment—nonetheless, OTTAWA would arguably align better with the three word sequence LOT OF WHAT. As we shall see, these potential misalignments are taken into account in the algorithm's mapping and comparison phases.

2.3 Stage 3: Mapping

The result of the previous phase is a series of alignments between the words in the reference text and those in a recognizer's hypothesis. In both of these texts there is named-entity (NE) markup. The next phase is to map the reference NEs to the hypothesis NEs. The result of this will be corresponding pairs of reference and hypothesis phrases, which will be compared for correctness in Stage 4.

Currently, the scorer uses a simple, greedy mapping algorithm to find corresponding NE pairs. Potential mapped pairs are those that overlap—that is, if some word(s) in a hypothesis NE have been aligned with some word(s) in a reference NE, the reference and hypothesis NEs may be mapped to one another. If more than one potential mapping is possible, this is currently resolved in simple left-to-right fashion: the first potential mapping pair is chosen. A more sophisticated algorithm, such as that used in the MUC scorer, will eventually be used that attempts to optimize the pairings, in order to give the best possible final score.

In the general case, there will be reference NEs that do not map to any hypothesis NE, and vice versa. As we shall see below, the unmapped reference NEs are completely missing from the hypothesis, and thus will correspond to recall errors. Similarly, unmapped hypothesis NEs are completely spurious: they will be scored as precision errors.

2.4 Stage 4: Comparison

Once the mapping phase has found pairs of reference-hypothesis NEs, these pairs are compared for correctness. As indicated above, we compare along three independent components: type, extent and content. The first two components correspond to MUC scoring and preserve backward compatibility. Thus our algorithm can be used to generate MUC-style NE scores, given two texts that differ only in annotation.

Type is the simplest of the three components: A hypothesis type is correct only if it is the same as the corresponding reference typer. Thus, in Example 4, hypothesis 1 has an incorrect type, while hypothesis 2 is correct.

Extent comparison makes further use of the information from the alignment phase. Strict extent comparison requires the first word of the hypothesis NE to align with the first word of the reference NE, and similarly for the last word. Thus, in Example 4, hypotheses 1 and 2 are correct in extent, while hypotheses 3 and 4 are not. Note that in hypotheses 2 and 4 the alignment phase has indicated a split between the single reference word GINGRICH and the two hypothesis words GOOD RICH (that is, there is a one- to two-word alignment). In contrast, hypothesis 3 shows the alignment produced by SCLite, which allows only one-to-one alignment. In this case, just as in Example 4, extent is judged to be incorrect, since the final words of the reference and hypothesis NEs do not align.

This strict extent comparison can be weakened by adjusting an *extent tolerance*. This is defined as the degree to which the first and/or last word of the hypothesis need not align exactly with the corresponding word of the reference NE. For example, if the extent tolerance is 1, then hypotheses 3 and 4 would both be correct in the extent component. The main reason for a non-zero tolerance is to allow for possible discrepancies in the lexeme alignment process—thus the tolerance only comes into play if there are word errors adjacent to the boundary in question (either the beginning or end of the NE). Here, because both GOOD and RICH are errors, hypotheses 3, 4 and 6 are given the benefit of the doubt when the extent tolerance is 1. For

```
Ref:  <P> NEWT      GINGRICH    </P>
Hyp1: <O> NEWT      GOODRICH    </O>
Hyp2: <P> NEWT      GOOD RICH   </P>
Hyp3: <P> NEWT        GOOD      RICH </P>
Hyp4: <P> NEWT    GOOD</P> RICH
Hyp5:     NEWT <P>  GINGRICH    </P>
Hyp6:     NEW  <P>  GINGRICH    </P>
```

Example 4

hypothesis 5, however, extent is judged to be incorrect, no matter what the extent tolerance is, due to the lack of word errors adjacent to the boundaries of the entity.

Content is the score component closest to the standard measures of word error. Using the word alignment information from the earlier phase, a region of intersection between the reference and the hypothesis text is computed, and there must be no word errors in this region. That is, each hypothesis word must align with exactly one reference word, and the two must be identical. The intuition behind using the intersection or overlap region is that otherwise extent errors would be penalized twice. Thus in hypothesis 6, even though NEWT is in the reference NE, the substitution error (NEW) does not count with respect to content comparison, because only the region containing GINGRICH is examined. Note that the extent tolerance described above is not used to determine the region of intersection.

Table 1 shows the score results for each of these score components on all six of the hypotheses in Example 4. The extent component is shown for two different thresholds, 0 and 1 (the latter being the default setting in our implementation).

2.5 Stage 5: Final Computation

After the mapped pairs are compared along all three components, a final score is computed. We use precision and recall, in order to distinguish between errors of commission (spurious responses) and those of omission (missing responses). For a particular pair of reference and hypothesis NE compared in the previous phase, each component that is incorrect is a substitution error, counting against both recall and precision, because a required reference element was missing, and a spurious hypothesis element was present.

Each of the reference NEs that was not mapped to a hypothesis NE in the mapping phase also contributes errors: one recall error for each score component missing from the hypothesis text. Similarly, an unmapped hypothesis NE is completely spurious, and thus contributes three precision errors: one for each of the score components. Finally, we combine the precision and recall scores into a balanced *F-measure*. This is a combination of precision and recall, such that F = 2PR / (P + R). F-measure is a single metric, a convenient way to compare systems or texts along one dimension[7].

[7]Because F-measure combines recall and precision, it effectively counts substitution errors twice. Makhoul et al. (1998) have proposed an alternate *slot error* metric that counts substitution errors only once.

Hyp	Type	Extent (0)	Extent (1)	Content
1	0	1	1	0
2	1	1	1	0
3	1	0	1	0
4	1	0	1	0
5	1	0	0	1
6	1	0	1	1

Table 1

3. Experiments and Results

To validate our scoring algorithm, we developed a small test set consisting of the Broadcast News development test for the 1996 HUB4 evaluation (Garofolo 97). The reference transcription (179,000 words) was manually annotated with NE information (6150 entities). We then performed a number of scoring experiments on two sets of transcription/NE hypotheses generated automatically from the same speech data. The first data that we scored was the result of a commonly available speech recognition system, which was then automatically tagged for NE by our system Alembic (Aberdeen 95). The second set of data that was scored was made availabe to us by BBN, and was the result of the BYBLOS speech recognizer and IdentiFinder™ NE extractor (Bikel 97, Kubala 97, 98). In both cases, the NE taggers were run on the reference transcription as well as the corresponding recognizer's output.

These data were scored using the original MUC scorer as well as our own scorer run in two modes: the three-component mode described above, with an extent threshold of 1, and a "MUC mode", intended to be backward-compatible with the MUC scorer.[8] We show the results in Table 2.

First, we note that when the underlying texts are identical, (columns A and I) our new scoring algorithm in MUC mode produces the same result as the MUC scorer. In normal mode, the scores for the reference text are, of course, higher, because there are no content errors. Not surprisingly, we note lower NE performance on recognizer output. Interestingly, for both the Alembic system (S+A) and the BBN system

[8]Our scorer is configurable in a variety of ways. In particular, the extent and content components can be combined into a single component, which is judged to be correct only if the individual extent and content are correct. In this mode, and with the extent threshold described above set to zero, the scorer effectively replicates the MUC algorithm.

Metric	Reference text		Recognizer output	
	A	I	S+A	B+I
Word correctness	1.00	1.00	0.47	0.80
MUC scorer	0.65	0.85	—	—
MITRE scorer (MUC mode)	0.65	0.85	0.40	0.71
MITRE scorer	0.75	0.91	0.43	0.76

Table 2

(B+I), the degradation is less than we might expect: given the recognizer word error rates shown, one might predict that the NE performance on recognizer output would be no better than the NE performance on the reference text times the word recognition rate. One might thus expect scores around 0.31 (i.e., 0.65×0.47) for the Alembic system and 0.68 (i.e., 0.85×0.80) for the BBN system. However, NE performance is well above these levels for both systems, in both scoring modes.

We also wished to determine how sensitive the NE score was to the alignment phase. To explore this, we compared the SCLite and phonetic alignment algorithms, run on the S+A data, with increasing levels of extent tolerance, as shown in Table 3. As we expected, the NE scores converged as the extent tolerance was relaxed. This suggests that in the case where a phonetic alignment algorithm is unavailable (as is currently the case for languages other than English), robust scoring results might still be achieved by relaxing the extent tolerance.

4. Conclusion

We have generalized the MUC text-based named entity scoring procedure to handle non-identical underlying texts. Our algorithm can also be used to score other kinds of non-embedded SGML mark-up, e.g., part-of-speech, word segmentation or noun- and verb-group. Despite its generality, the algorithm is backward-compatible with the original MUC algorithm.

The distinction made by the algorithm between extent and content allows speech understanding systems to achieve a partial score on the basis of identifying a region as containing a name, even if the recognizer is unable to correctly identify the content words. Encouraging this sort of partial correctness is important because it allows for applications that might, for example, index radio or video broadcasts using named entities, allowing a user to replay a particular region in order to listen to the corresponding content. This flexibility also makes it possible to explore information sources such as prosodics for identifying regions of interest even when it may be difficult to achieve a completely correct transcript, e.g., due to novel words.

Extent Tolerance	SC-Lite Alignment	Phonetic Alignment
1	0.42	0.43
2	0.44	0.45
3	0.45	0.45

Table 3

Acknowledgements

Our thanks go to BBN/GTE for providing comparative data for the experiments duscussed in Section 3, as well as fruitful discussion of the issues involved in speech understanding metrics.

References

J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson , M. Vilain (1995). "MITRE: Description of the Alembic System as Used for MUC-6", in *Proceedings of the Sixth Message Understanding Conference.*

D. Bikel, S. Miller, R. Schwartz, R. Weischedel (1997). "NYMBLE: A High-Performance Learning Name-finder", in *Proceedings of the Fifth Conference on Applied Natural Language Processing.*

N. Chinchor (1995). "MUC-5 Evaluation Metrics", in *Proceedings of the Fifth Message Understanding Conference.*

W.M. Fisher, J.G. Fiscus (1993). "Better Alignment Procedures for Speech Recognition Evaluation". *ICASSP* Vol. II.

J. Garofolo, J. Fiscus, W. Fisher (1997) "Design and Preparation of the 1996 Hub-4 Broadcast News Benchmark Test Corpora", in *Proceedings of the 1997 DARPA Speech Recognition Workshop.*

F. Kubala, H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, J. Makhoul (1997) "The 1996 BBN Byblos Hub-4 Transcription System", in *Proceedings of the 1997 DARPA Speech Recognition Workshop.*

F. Kubala, R. Schwartz, R. Stone, R. Weischedel (1998) "Named Entity Extraction from Speech", in *Proceedings of the Broadcast News Transcription and Understanding Workshop.*

J. Makhoul, F. Kubala, R. Schwartz (1998) "Performance Measures for Information Extraction". unpublished manuscript, BBN Technologies, GTE Internetworking.

M. Marcus, S. Santorini, M. Marcinkiewicz (1993) "Building a large annotated corpus of English: the Penn Treebank", *Computational Linguistics*, 19(2).

R. Merchant, M. Okurowski (1996) "The Multilingual Entity Task (MET) Overview", in *Proceedings of TIPSTER Text Program (Phase II).*

G.R. Sampson (1995) *English for the Computer*, Oxford University Press.