Experiments in Multi-Modal Automatic Content Extraction

Lance Ramshaw, Elizabeth Boschee, Sergey Bratus, Scott Miller,
Rebecca Stone, Ralph Weischedel, and Alex Zamanian

BBN Technologies
70 Fawcett St.
Cambridge, MA 02138 USA
1-617-873-2236

{lramshaw, eboschee, sbratus, szmiller, rwstone, weischedel, azamania}@bbn.com

ABSTRACT

Unlike earlier information extraction research programs, the new ACE (Automatic Content Extraction) program calls for entity extraction by identifying and linking all of the mentions of an entity in the source text, including names, descriptions, and pronouns. Coreference is therefore a key component. BBN has developed statistical co-reference models for this task, including one for pronoun co-reference that we describe here in some detail. In addition, ACE calls for extraction not just from clean text, but also from noisy speech and OCR input. Since speech recognizer output includes neither case nor punctuation, we have extended our statistical parser to perform sentence breaking integrated with parsing in a probabilistic model.

1. INTRODUCTION

The Automatic Content Extraction (ACE) program, a new effort to stimulate and benchmark research in information extraction, presents two challenges:

1. *Recognition of entities is paramount.* In named entity evaluations, recognizing and classifying name strings is the focus; in the MUC Template Element (TE) task, all names for an entity but only one description were to be collected. In the ACE entity detection and tracking (EDT) task, all mentions of an entity, whether a name, a description, or a pronoun, are to be found and collected into equivalence classes based on reference to the same entity. Therefore, practical co-reference resolution is fundamental.
2. *Extraction is measured not merely on text, but also on speech and on OCR input.* Named entity recognition had previously been benchmarked on text, speech, and OCR, but extraction above the level of names had rarely been attempted. Moving beyond name finding is a crucial leap for modalities other than text, since the ability to relate two strings (as in ACE) in very noisy input may degrade much more than finding strings in isolation (as in named entity recognition.) Furthermore, the lack of case and punctuation, including the lack of sentence boundary markers, poses a challenge to full parsing of speech.

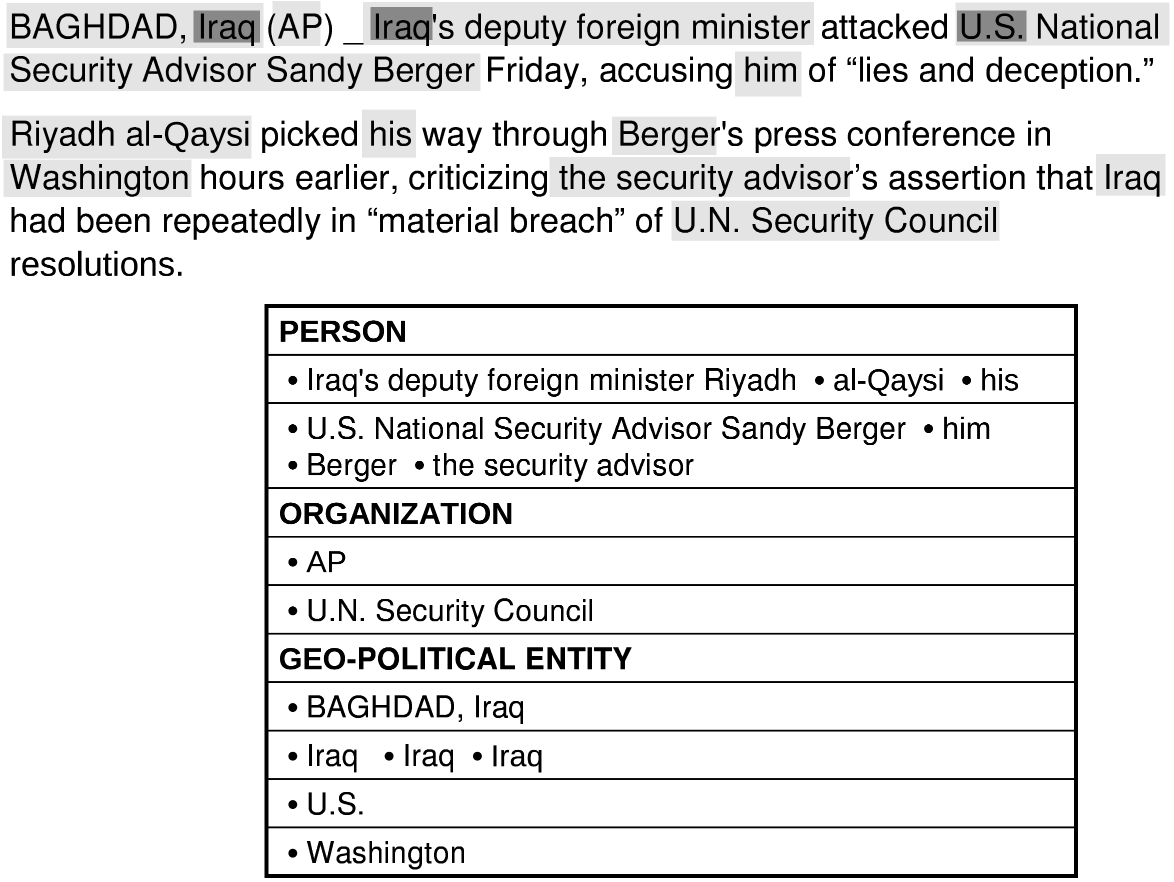
To address challenge 1 above, BBN developed statistical learning algorithms for pronoun resolution and name co-reference and is developing a statistical learning algorithm for co-reference of definite noun phrases (beyond names and pronouns). The pronoun co-reference algorithm is described here.

Challenge 2 did not require abandoning our statistical approach to full parsing, even though there is no punctuation in automatic speech recognition (ASR) output, which removes many of the clues that help to determine sentence boundaries in printed text. Rather, we developed a technique to parse a window of words, successively sliding the window a word at a time over a whole speaker turn. A non-overlapping sequence of trees that covers the speaker turn is chosen to obtain full parses of ASR output. As a side effect of selecting full parses for a speaker turn, sentence boundaries are predicted. This new algorithm is described here.

In addition to describing these two algorithms, this paper overviews the task briefly, describes the system, and reports results from two evaluations performed under the auspices of NIST.

2. TASK

The ACE program uses the term “mention” for any text span that refers to an entity of one of the ACE target types. For example, in the phrase “Lincoln was 51 when he became president of the US”, “Lincoln” is a name mention, “he” is a pronoun mention, and “president of the US” is a nominal (other noun phrase) mention. In the current specification for the ACE Entity Detection and Tracking (EDT) task, all mentions of an entity are to be collected within a document. The entity must be classified by type, i.e., person, organization, location, facility, or GPE (geo-political entity: country, state, province, or city). In addition to the “type” attribute, all names, if any, are reported as “name” attributes for the entity. Future versions of the task specification may include both additional types of entities and additional attributes for each entity, and will include tracking entities across documents, rather than merely within documents.



BAGHDAD, Iraq (AP) _ Iraq's deputy foreign minister attacked U.S. National Security Advisor Sandy Berger Friday, accusing him of “lies and deception.”

Riyadh al-Qaysi picked his way through Berger's press conference in Washington hours earlier, criticizing the security advisor’s assertion that Iraq had been repeatedly in “material breach” of U.N. Security Council resolutions.

PERSON
• Iraq's deputy foreign minister Riyadh • al-Qaysi • his
• U.S. National Security Advisor Sandy Berger • him • Berger • the security advisor
ORGANIZATION
• AP
• U.N. Security Council
GEO-POLITICAL ENTITY
• BAGHDAD, Iraq
• Iraq • Iraq • Iraq
• U.S.
• Washington

Figure 1: Sample Text with EDT Entities and Mentions

Figure 1 shows a sample of text with the mentions of EDT entities highlighted, and a table showing the types of EDT entities and listing the different mentions for each.

3. BRIEF SYSTEM OVERVIEW

BBN’s ACE system for the EDT task involves three primary components: name finding, parsing, and co-reference. The name finding component [1] provides some of the strongest clues for entity detection and tracking. The parsing component [2] determines the extent and head word of each mention, which is particularly useful for those noun phrases not headed by proper names. Both components are implemented as trained statistical models. The parsing model considers only parses that are consistent with the name boundaries already predicted by the name finding model.

There are co-reference components for names, for pronouns, and for other noun phrases. For names, the model decides for each name mention encountered whether it is more likely to be the first mention of a new entity or if it should be linked to a previous name mention of some existing entity. For pronouns, the model determines similarly for each pronoun mention which earlier mention (whether pronoun, name, or nominal) it should be linked to, or whether it should be left unlinked. The nominal co-reference component performs two tasks for every noun phrase in the parse:

1. Determine what ACE class, if any, the noun phrase has.
2. Determine which previous entity the noun phrase refers to or that this is an entity not previously seen in the document.

BBN has statistical models for all of these tasks, though the nominal co-reference model was not ready in time for the formal evaluation in early November. The following section describes the pronoun co-reference model in more detail.

4. PRONOUN CO-REFERENCE MODEL

A statistical model is used to predict pronoun co-reference. Although the algorithm is designed to produce antecedents for all pronouns except expletives and those with implicit antecedents, our focus was on cases when the antecedent was an ACE mention. We could therefore focus on connecting the parse constituent corresponding to a pronoun either to an NP in a parse tree or else declaring the pronoun “unresolvable” when no such constituent node could be found. The pronoun resolution algorithm takes as input a parse tree where each constituent corresponding to a mention has been labeled with one of the EDT types (*Person, Organization, GPE, Location,* or *Facility*) and with the mention type (*Name* or *Descriptor*). Further, if the mention has already been found to be a member of a co-reference chain by the name or nominal co-reference components, the constituent node was also labeled with the ID of this chain.

Pronouns are processed in a depth-first traversal of the parse tree. For each pronoun, all earlier NP nodes in the document are considered as possible antecedents. Candidates are processed by walking backwards through the parse trees from the pronoun towards the beginning of the document (as proposed by Hobbs [3]). Each of these NPs and the "unresolvable" case are then scored using the following model, and the choice with the highest probability is selected.

The goal of the probabilistic model is choose the antecedent (*ant*) so as to maximize its probability given the pronoun (*pro*) and its surrounding environment (*env*). Using Bayes Rule to invert the probabilities and an independence assumption to separate the pronoun from its environment, this is approximately equivalent to the following expression:

$$P(ant \mid pro, env) \approx \frac{P(ant)P(pro \mid ant)P(env \mid ant)}{P(pro, env)}$$

Since the denominator is constant regardless of the choice of antecedent, we only need to maximize the following expression:

$$P(ant)P(pro \mid ant)P(env \mid ant)$$

As features to predict the probability of the possible antecedents, we use their *Type* (either one of the EDT types or *Undetermined* for non-mention candidates), their *Number* and *Gender*, and their *Distance*. The distance for an antecedent is computed by searching through the parse trees of the current and previous sentences in the order suggested by Hobbs [3], and counting the number of NP constituents between the pronoun and the antecedent. Making another independence assumption, the distance is also modeled separately from the type, number, and gender.

For example, in the following case:

… Mrs. Brown … < 7 other constituents > … She said …

the probability of the "Mrs. Brown" phrase being the antecedent is computed as follows:

$$P(ant) \approx P(person, singular, female)\ P(dist=7)$$

The probability of the pronoun itself is estimated as the probability of that particular word, conditioned on the type, number, and gender of the antecedent, in this case:

$$P(pro \mid ant) \approx P(\text{“she”} \mid person, singular, female)$$

As a feature for estimating the probability of the pronoun's environment, we used just the head word of the constituent that was the parent of the pronoun in the parse tree, conditioned on the type of the antecedent. In this example, that means:

$$P(env \mid ant) \approx P(\text{“said”} \mid person)$$

When the parent head word in the actual parse tree was an auxiliary verb, it was augmented by the main verb.

The training counts for each part of the model were taken from 300K words of Penn Treebank data that had been annotated for pronoun co-reference. The lexical model for the parent head word was smoothed by using the uniform distribution as a back-off.

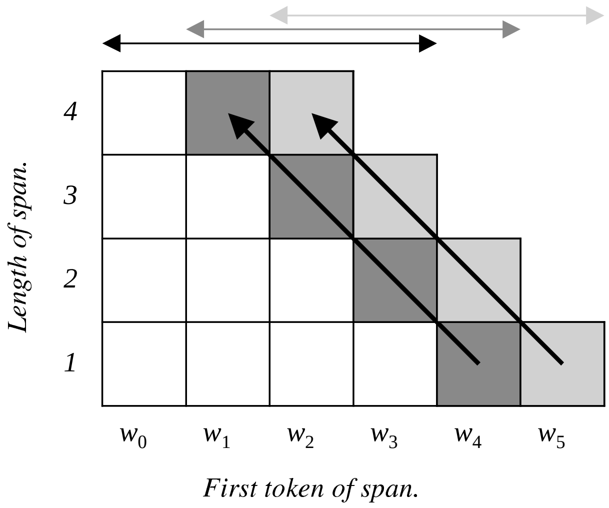


Figure 2: Windowing Parser, window_size=4

5. PARSING AUTOMATICALLY TRANSCRIBED SPEECH

For speech, in order to apply our system to ASR output, we modified the parser component to combine syntactic parsing and sentence-breaking functionality into a single module. Our primary goal in integrating these two processes was to avoid the serious parsing errors that could be caused by relying on a potentially errorful independent sentence-breaking mechanism. At the same time, we wanted as much as possible to maintain optimal parser accuracy.

For each speaker turn, we begin by providing the parser with a window of the first N words of text. The bottom-up, statistical parser is then called to construct a chart for that initial portion of the text, showing the syntactic constituents that can cover each span of words, along with their estimated probabilities. Some of those chart cells will typically contain "TOP" entries, the symbol used for separate utterances in the training data (which syntactically are often sentences, but can also be isolated noun phrases or the like).

The algorithm then shifts the chart window one word to the right, giving the parser words 2 through N+1, computing any new constituents that include the new word. The process of sliding the window over proceeds until the parser has processed the last N words of the speaker turn. The window size N defines the maximum number of tokens per constituent (and thus the maximum sentence length). This value is set at 30 in the current system. The final parse of the speaker turn is then formed by finding the sequence of sentence parses that exactly covers the full text with the best overall probability.

At the lowest levels, the windowing chart algorithm runs much the same as the original, updating chart entries from the same subordinate entries and according to identical statistical formulas. The differences lie in the "outer loops" of the algorithm.

1. First, rather than filling in the chart in the usual left-to-right, bottom-to-top order, we iterate over tokens, proceeding upward along the diagonal for each [see Figure 2].

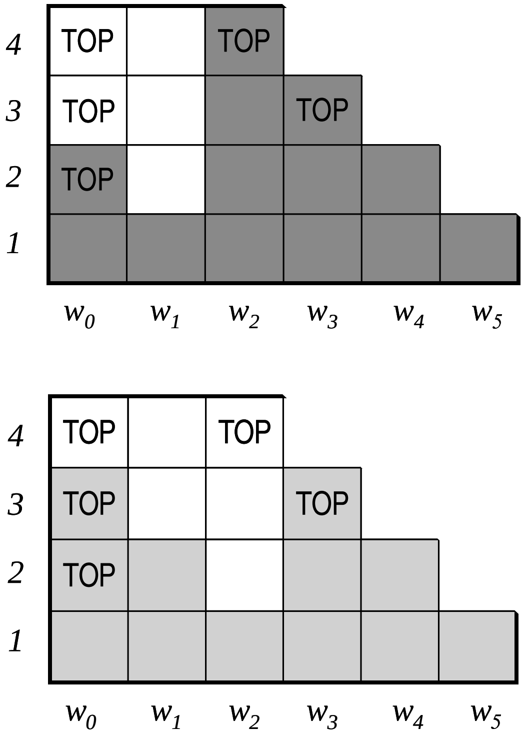

4
3
2
1
TOP
TOP
TOP
TOP
TOP
w_0 w_1 w_2 w_3 w_4 w_5
4
3
2
1
TOP
TOP
TOP
TOP
TOP
w_0 w_1 w_2 w_3 w_4 w_5

Figure 3: Alternate Possible Sentence Sequences

2. We cap our chart at window_size rows, thereby constraining the sentence length and the required parse computation. Only window_size columns are active at a time, making it simple to visualize shifting the chart by one column per token iteration.
3. Unlike the standard parser, which at the end of each sentence only has to choose the best sentence (TOP) constituent from the upper-left-most chart entry, this algorithm needs to select a sequence of high probability sentence (TOP) constituents that together cover the entire speaker turn. We use a Viterbi search to determine the best path through this space of sentences.

At the end of processing a speaker turn, the algorithm must search back to identify the most likely sequence of TOP constituents that together cover all the words in the turn. For example, given the TOP constituents found in Figure 3, that six-word turn could be parsed as shown either as a two-word sentence followed by a four-word sentence, or as two three-word sentences. The four-word TOP constituent beginning at word 0 does not form an alternative path because no TOP constituent covers words 4 and 5. (In the actual system, there is a fall-back provision allowing any individual words to be treated as a TOP constituent. This ensures that some path can always be found, although an artificial probability cost is added for each word so treated that is high enough to force the system to prefer linking normal TOP constituents whenever possible.)

Using the Viterbi algorithm to efficiently search for the most likely sequence involves storing for each column the cost of the best path up to that point and a pointer to the final constituent along that best path. The search proceeds first from left to right. At each position, there are at most window_size possibilities to consider, working up the diagonal. If a TOP element is found along that diagonal in column *k*, the cost of the best path through that element is the cost of the element itself combined with the known cost of the best path up to column *k*-1. Once the optimal cost for a path through the entire speaker turn has been computed, the final constituent pointers can be traced back right-to-left to output the complete best path.

6. PERFORMANCE

Two ACE EDT evaluations were performed in 2000, with four participating sites submitting systems. Extensive graphs analyzing the combined results are available through the NIST Web site[1]. Table 1 shows the entity error rates from the second of these evaluations for BBN's system when run both on newswire texts and on the ASR output from broadcast news programs.

The scoring program searched for the mapping between the entities found by the system and those in the answer key that best aligned their mentions. Based on that mapping, answer key entities not found by the system were counted as misses, and system entities not in the answer key were counted as false alarms. Mapped entities to which the system had assigned the wrong type were counted as errors. The final column sums the three kinds of error.

These scores were state-of-the-art as of the November evaluation. Human performance based on limited studies of inter-annotator agreement is estimated at roughly 20% sum of errors.

Table 1: BBN Entity Detection Results

Entity Scores	Miss	False Alarm	Error	Sum of Errors
Newswire	28.2	24.9	8.3	61.4
Broadcast News ASR	38.9	28.6	6.4	73.9

7. REFERENCES

[1] Bikel, D., Schwartz, R., and Weischedel, R., "An Algorithm that Learns What's in a Name," Machine Learning 34 (1999), 211-231,

[2] Miller, S., Ramshaw, L., Fox, H., and Weischedel, R. "A Novel Use of Statistical Parsing to Extract Information from Text", In Proceedings of 1st Meeting of the North American Chapter of the ACL, (Seattle, WA, 2000), 226-233.

[3] Hobbs, J. R., "Resolving Pronoun References", reprinted in 1986 in Readings in Natural Language Processing, B. Grosz, K. Jones, and B. Webber, eds., Morgan Kaufmann, (1977)

[1] http://www.itl.nist.gov/iad/894.01/tests/ace/phase1/workshop.htm and ftp://jaguar.ncsl.nist.gov/ace/phase1/acekick/nist-2000-11-edt-results.pdf