HUB-4 "Templette" Task Scoring Procedure, Version 0.3

Introduction

This document describes how the *reference* (human generated data) will be compared to the *hypothesis* (system generated data) in the HUB-4 "templette" task. The scoring procedure assumes that a two-step information extraction method will be used to produce both the reference and the hypothesis.

Step 1: Source Documents to Text Documents

In the first step, the *source corpus* is converted to a *text corpus*. The source corpus is a set of *source documents*. A source document may be a newswire article in ASCII text format, an audio or video file, a printed page, or something else. The exact definition may be found in the test procedures. The text corpus is one computer file, consisting of a sequence of *text documents*. The format of text documents is described below. The nature of this conversion depends on the format of the source corpus. The conversion may be trivial if the source corpus is already ASCII text, or may be non-trivial (e.g., if the source corpus is a set of audio files).

Step 2: Text Documents to Template Sets

In the second step of the assumed pipeline, information from the text corpus is extracted and organized into a *template set*. A template set is one computer file. The format of template sets is detailed below.

The scoring procedure requires the reference and hypothesis template sets as input. If the reference and hypothesis text corpora differ, then they will also be required.

Scoring Input Formats

Text Corpus Format

As mentioned above, a *text corpus* is a single computer file, consisting of a sequence of *text documents*. A pair of SGML tags, called the *document tags*, encloses each text document. Often the generic identifier for document tags is "DOC", but any generic identifiers may be used. All document tags in a text corpus should have the same generic identifier.

Each text document should have an SGML *document identification element* whose contents uniquely identify the text document in the text corpus. Often the generic identifier for document identification elements is "DOCNO" or "DOCID", but any generic identifier may be used (all document identification elements in a text corpus should use the same generic identifier).

The contents of the document identification element are used to associate information in an instance set with a text document, and also to associate information in a reference instance set with information in a hypothesis instance set.

The appendix contains a hypothetical text corpus, consisting of three text documents. The document tags are named "DOC", and the document identifier tags are named "DOCNO". It may be helpful to refer to the examples in the appendix in the following sections.

Format of Extracted Information

The appendix also contains hypothetical reference and hypothesis template sets, made of information extracted from the text document.

The information extracted is contained in a single computer file. The information has a hierarchical structure. From top to bottom, the levels of the hierarchy are as follows:

- Template Set
- Instance Set
- Instance
- Slot
- Fill-Alternative
- Single Fill

(A discussion of the terminology differences between this document and the general guidelines [HIRSCHMAN_HUB4] follows these definitions.)

Template Sets

At the top of the hierarchy, the entire file is a *template set*. A template set is a sequence of *instance sets*. If a text document is deemed to contain a reportable story, exactly one instance set should be created for that text document. If the document does not contain a reportable story, an "empty" instance set may be created, but this is not necessary for scoring.

Our example has three text documents in the text corpus. Only the last document was deemed relevant, and in our example the first two instance sets are "empty".

Instance Set

An *instance set* is a collection of *instances* created from one text document. In our example, the last instance set of the template set is:

```
<TEMPLATE-PRI19980302.2000.2923-1> :=
    DOC_NR: PRI19980302.2000.2923 ##14#35#
    EVENT: <SPORTS_EVENT-PRI19980302.2000.2923-1>
    COMMENT: "No locations for earlier tournaments."
<SPORTS_EVENT-PRI19980302.2000.2923-1> :=
    S_EVENT: "African cup of nation soccer tournament" ##216#255#
           / "the African cup" ##401#416#
           / "the tournament" ##461#475#
    WINNER: "Egypt" ##332#337#
    LOSER: "defending champion [south Africa]" ##295#326#314#326#
    SCORE: "2-0" ##327#330#
    LOCATION: "south Africa" ##314#326#
            / "The host of the tournament" ##449#475#
    DATE: "03/02/1998" ##89#99#
```

```
    COMMENT: "location of earlier tournaments unstated"
```

Instances

The instance set in the previous section consists of two instances, one of type `TEMPLATE`, and one of type `SPORTS_EVENT`.

Each instance must contain a *header* and a *body*. The header consists of an *instance pointer*, followed by the string "`:=`" on the same line. For example, this is an instance header:

```
<SPORTS_EVENT-PRI19980302.2000.2923-1> :=
```

The body of an instance is a set of slots.

One slot in an instance may serve to mark the instance as *optional*. For instance, the above SPORTS_EVENT instance could be marked optional with an OBJ_STATUS slot:

```
<SPORTS_EVENT-PRI19980302.2000.2923-1> :=
    S_EVENT:    ...
    WINNER:     ...
    LOSER:      ...
    SCORE:      ...
    LOCATION:   ...
    DATE:       ...
    OBJ_STATUS: OPTIONAL
    COMMENT:    ...
```

Slots

There are eight slots In the above instance, named `S_EVENT, WINNER, LOSER, SCORE, LOCATION, DATE, OBJ_STATUS` and `COMMENT`. Each slot consists of a *slot name* and a *slot body*. The slot name is a string of letters, numbers, hyphens or underscore characters, followed by a colon. It identifies the slot within the instance. The slot body is a set of *fill-alternatives*. Hypothesis slots may have only one fill-alternative. Reference slots may have several fill-alternatives. Fill-alternatives in a reference slot are separated by slash characters as the first non-blank character on a line.

Fill-Alternatives

A fill-alternative is a set of *single fills*. Our example has no fill-alternatives consisting of more than one single fill, but such a fill-alternative is possible. For instance, if a SPORTS_EVENT instance was to be created for the 1986 African Cup, then the fill-alternative in the EVENT slot of the top-level TEMPLATE instance would contain two single fills of type pointer.

```
<TEMPLATE-PRI19980302.2000.2923-1> :=
    DOC_NR: PRI19980302.2000.2923 ##14#35#
    EVENT: <SPORTS_EVENT-PRI19980302.2000.2923-1>
           <SPORTS_EVENT-PRI19980302.2000.2923-2>
    COMMENT: "1986 tournament optional, since no location given"
```

(The instance `<SPORTS_EVENT-PRI19980302.2000.2923-2>` in the above EVENT slot would be marked optional with the OBJ_STATUS slot.)

The current task description says that only the EVENT slot of the TEMPLATE instance may have more than one single fill in a fill-alternative.

Single Fills

For the current task, there are two types of single fills:

- Pointer Fills
- Text Fills

A single fill of either type must be on its own line.

Pointer Fills

Pointer fills refer to instances in a template set. The pointer format is used in both pointer fills and in the pointer part of instance headers. Here is a pointer in our example:

```
<SPORTS_EVENT-PRI19980302.2000.2923-1>
```

The entire pointer is enclosed in angle brackets, and consists of three character strings, separated by hyphens. The first string is the *instance type*, as defined in the task guidelines. The second string is the *document identifier*. The document identifier should be identical to the non-blank characters of the document identification element in the text document. The third string in a pointer is the instance's *one-up* number. It is used to identify the instance within the instance set. No two instances of the same type in the same instance set should have the same one-up number.

Text Fills

Here is a text fill from our example:

```
  "defending champion [south Africa]" ##295#326#314#326#
```

Each *text fill* consists of two parts, the *content part* and the *extent part*.

Content Part

The content part of a text fill is a *content string*, optionally enclosed in double quotes. A content string is a character string copied from the text document. Content strings in the reference may have pairs of square brackets inserted in them, indicating *minimal content strings*, which are substrings of the *maximal content string*. All newline characters in a content string must be converted to space characters when the content string is copied into the text fill. This is the content part of the above text fill:

```
"defending champion [south Africa]"
```

This content part consists of the maximal content string

```
"defending champion south Africa"
```

and the minimal content string

```
"south Africa"
```

Extent Part

The extent part of a text fill is a sequence of pairs of numbers. Both the pairs and the numbers within the pairs are separated by single hash characters. The extent part begins with double hash characters. This is the extent part of the above text fill:

```
##295#326#314#326#
```

There should be no spaces in the extent part. Each pair of numbers is called an *extent*. Extents specify the location of the beginning and ending of a piece of information in a document.

It should be noted that what the numbers within extents actually represent varies. In the inputs to the current scoring procedure they are byte offsets (explained below). During scoring, they are transformed in an "extent normalization" step into units based on the phonetic alignment of the reference and hypothesis texts. In future scoring procedures, extents could possibly refer to the start and end times of a part of an audio recording.

The first integer of each extent (called the *start offset*) is the character index in the text document of the first character in a (maximal or minimal) content string. The second integer (the *end offset*) is the character index in the text document of the first character *following* the fill. Character indices are calculated by counting characters, starting at zero with the "<" character in the document tag (in our example, the document tag is "<DOC>").

One extent `(a, b)` is said to *enclose* another extent `(c, d)` iff

```
a <= c <= b
```

and

```
a <= d <= b
```

One extent `(a, b)` is said to *overlap* another extent `(c, d)` iff

```
a <= c <= b
```

or

```
a <= d <= b
```

or

```
c <= a <= d
```

or

```
c <= b <= d
```

The first extent in the extent part of a fill gives the locations of the beginning and ending of the fill's maximal content string. In a reference fill, any extents after the first one give the locations of minimal content strings.

Notes on the above Terminology

Because this document is concerned more with the mechanics of comparing reference and hypothesis and less with the meaning of the current task, different terms are used to avoid clashes.

"Templette/Template" Versus "Instance/Instance Set"

The term "instance" refers to a "tuple" created in the extraction process. For the current task, there are both "templette" instances and "template" instances. A templette instance is created for each "event" of the general guidelines. To organize all the templette instances from one text document into a single structure, one template instance is also created. The set of all templette instances from one text document, together with the single "tie-up" template instance, is what we refer to as an instance set.

"Fill-alternative" Versus "Multiple-fill"

Some earlier scoring documentation [DOUTHAT_MESSAGE] used the term "multiple fill" to refer to what is called a "fill-alternative" in this document. The earlier term didn't capture the sense of "alternative-ness." In the current task it would be even more confusing, since the valence of all slots but the EVENT slot is one. Therefore, the current term has replaced the old one.

Scoring Algorithm

Inputs

The scoring algorithm takes as input the reference and hypothesis template sets. In addition, if the text corpus used by the hypothesis is different from the text corpus used by the reference, the reference and hypothesis text corpora are also required.

Outputs

The scoring algorithm produces as output a *mapping* of the reference template set (and its sub-structures) to the hypothesis template set (and its sub-structures). Various metrics will be applied to the mapping to produce measurements of the similarity of the hypothesis to the reference. The most prominent metric used is the *F-measure*, a function of the number of correctly mapped single fills in the reference and hypothesis. These similarity measurements, or *scores*, will also be output.

Steps in the Scoring Algorithm

Normalization

Content Normalization

If the hypothesis and reference texts differ, valid spelling differences will be removed by transforming non-normal (but valid) word spellings into normal ones. This will be done by means of a *Global Mapping File*, developed by NIST for use in their evaluations of automatic speech recognition systems [NIST_HUB4]. A global mapping file specifies a list of words that have a non-normal spelling, and for each word the normalized spelling.

Extent Normalization by Alignment of Text Corpora

If the reference and hypothesis text corpora differ, the extents in text fills will not be immediately comparable. If this is the case, the extents will be *normalized*, by first aligning the two corpora using a dynamic programming algorithm, then recalculating the extents based on the alignment [MITRE_MSCORE, BURGER_NAMED].

Example Alignment

To illustrate extent normalization, here is a simplified example of a pair of different text documents from the same source document, with the character extent indices shown below the texts:

```
REF:  <DOC> The h- heart of General Motors </DOC>
HYP:  <DOC> A part of General Motors       </DOC>
NDX:  0123456789012345678901234567890123456
```

And here is a pair of text fills from the text documents:

```
REF:  "General Motors" ##22#36#
HYP:  "General Motors" ##16#30#
```

(For brevity, the document identifier has been left out of the texts and the fills.) It can be seen that the extents in the text fills do not agree when based on character count.

If a phonetic alignment program [FISHER_TALD3E, PICONE_AUTO, FISHER_BETTER, FISHER_FURTHER] is used to align the two texts, the resulting alignment would look something like this:

```
REF:  | THE | H- | HEART | of | general | motors |
HYP:  | A   |    | PART  | of | general | motors |
NDX:  0     1    2       3    4         5        6
```

The normalized indices are calculated by counting the vertical bars produced by the alignment, rather than the number of characters. Rewriting the text fills with the normalized indices, we obtain

```
REF:  "General Motors" ##4#5#
HYP:  "General Motors" ##4#5#
```

and can see that the normalized extents do in fact match.

Removal of Non-evaluated Text

Before mapping the reference and hypothesis instance sets, premodifiers (e.g., the words "and", "a", and "the") will be removed from all fills and the corresponding extents adjusted.

Other substrings, such as certain punctuation marks, may be *whited out*: the unwanted text will be changed to whitespace, but the extents will not be adjusted. (When fills are compared for content, each whitespace string is changed to a single space character).

It is possible that a fill may begin and end in the same extractable SGML section, but include some non-extractable SGML section, as in the following example:

```
TEXT:

And one final sports note -- today in
Anchorage,...
<ANNOTATION>  (voice-over)  </ANNOTATION>
...Alaska, the ceremonial start of the 26th

FILL:

"Anchorage,... <ANNOTATION>  (voice-over)  </ANNOTATION> ...Alaska"
```

When this is the case, the non-extractable section will be whited out. The above example would then look like this (depending on what punctuation is removed):

```
WHITED-OUT FILL:

"Anchorage                                                  Alaska"
```

Mapping

The *mapping* of hypothesis template sets to reference template sets is the association of structures in the two template set hierarchies [CHINCHOR_FOUR]. Each structure in the hypothesis hierarchy is associated with at most one structure in the reference hierarchy and vice versa.

One structure may only be associated with another structure from the same level in the hierarchy: instance sets with instance sets, instances with instances, slots with slots, etc.

Further, if structure A is mapped to structure B, structure A's child structures may only map to structure B's child structures. For instance, if one instance is mapped to another instance, the first instance's slots may only be mapped to the other instance's slots.

To determine mappings of structures at several levels of the hierarchy, *points* are used. Points are categorized as correct, incorrect, missing, spurious, or unscored. At the bottom of the structure hierarchy, each mapping of either a single fill to another single fill or of a single fill to nothing results in one or two points. At any level in the structure hierarchy above the single fill, the points resulting from the mapping of one structure to another are determined by combining the points from the structures at the next lower level. For instance, the points obtained from mapping one instance to another are the sum of the points from the mappings of each of the slots of the instances (ignoring unscored slots). The ways points are combined at each level of the mapping are described below.

The following simple greedy algorithm is used to map objects at several levels in the hierarchy:

General Greedy Mapping Algorithm

1. generate all possible `(reference-object, hypothesis-object)` pairs. For each pair, calculate the points, and from the points the pair's F-measure.
2. Put the pair with the highest F-measure in the list of chosen mappings. Remove from consideration all pairs which contain either of the objects of the selected pair.
3. Repeat the previous step until there are no pairs left under consideration.

Mapping Template Sets

At the very top of the structure hierarchy, the mapping of the single reference template set to the single hypothesis template set is trivial. The points from the template set mapping are the sum of the points from each instance set mapping.

Mapping Instance Sets

Segmented Hypothesis Documents

At the "instance set" level, a reference instance set is mapped to a hypothesis instance set based only on the document identifier used in the pointers of the instance sets' instance headers.

For example if the instance header lines from the reference template set are:

```
<TEMPLATE-VOA19980126.2100.1446-1> :=
<BREAD-VOA19980126.2100.1446-1> :=
<CIRCUS-VOA19980126.2100.1446-1> :=
<CIRCUS-VOA19980126.2100.1446-2> :=

<TEMPLATE-VOA19980302.1600.0096-1> :=

<TEMPLATE-VOA19980111.2300.0414-1> :=
<BREAD-VOA19980111.2300.0414-1> :=
<BREAD-VOA19980111.2300.0414-2> :=
<CIRCUS-VOA19980111.2300.0414-1> :=
```

and those from the hypothesis template set are:

```
<TEMPLATE-VOA19980126.2100.1446-1> :=
<BREAD-VOA19980126.2100.1446-1> :=
<BREAD-VOA19980126.2100.1446-2> :=
<CIRCUS-VOA19980126.2100.1446-2> :=

<TEMPLATE-VOA19980302.1600.0096-1> :=
<CIRCUS-VOA19980302.1600.0096-1> :=

<TEMPLATE-VOA19980111.2300.0414-1> :=
<CIRCUS-VOA19980111.2300.0414-1> :=
<CIRCUS-VOA19980111.2300.0414-2> :=
<CIRCUS-VOA19980111.2300.0414-3> :=
```

then at the instance set level, the mappings will be (not surprisingly):

```
VOA19980126.2100.1446  <---> VOA19980126.2100.1446
VOA19980302.1600.0096  <---> VOA19980302.1600.0096
VOA19980111.2300.0414  <---> VOA19980111.2300.0414
```

When one instance set is mapped to another, the points for the mapping are just the sum of the points from each instance mapping.

Unsegmented Hypothesis Texts

It is possible that the text corpus used to make the hypothesis will not be segmented into text documents. For instance, the hypothesis might be referring to the raw output of an automatic speech recognizer, which consists only of a list of words and SGML timestamp tags.

If this is the case, each *hypothesis instance set extent* will be defined as a pair `(a, b)`, where `a` is the smallest offset of all text fills in a hypothesis instance set, and `b` is the largest offset of all text fills in the instance set.

The reference instance sets will always be based on segmented data. The *reference instance set extent* will consist of the smallest and largest possible offsets of the text document. When working with unsegmented hypothesis data, offsets will be measured relative to the beginning of the entire text corpus, rather than the beginning of each text document.

Two instance sets overlap if their instance set extents overlap.

With unsegmented hypothesis texts, a modified form of the general greedy algorithm is used to map instance sets. The modification is the restriction that only instance sets which overlap may be mapped.

It can be shown that when the hypothesis text corpus is segmented the same way as the reference text corpus, and the hypothesis instance sets respect the segmentation, the two algorithms for mapping instance sets give the same results.

Mapping Instances

At the instance level, a reference instance may be mapped to a hypothesis instance only if the two are of the same type. The points for the mapping are the sums of the points from the mappings of the scored slots in the two instances. Slots which are specified as "unscored" do not contribute to the points of the instance mapping.

The above greedy algorithm is used to map reference instances of a single type to hypothesis templates of the same type.

Mapping Slots

The slots of a reference instance are mapped to the slots of a hypothesis instance by slot name. The points of a slot mapping are the sum of the points from the slots' fill-alternative mapping.

Mapping Fill-Alternatives

The one fill-alternative in a hypothesis slot is mapped to whichever of the fill-alternatives in the reference slot gives the best F-measure. Any leftover fill-alternatives in the reference are mapped to nothing and do not contribute any points.

Mapping Single Fills

The single fills in a hypothesis fill-alternative are mapped to the single fills in the reference fill-alternative using the general mapping greedy algorithm. The points from a single fill mapping are calculated based on the type of the single fill.

Comparison of Pointer Fills

The mapping of one pointer fill to another gives one point. The point is determined as follows:

Correct
: The instance referred to by the reference pointer fill is mapped to the instance referred to by the hypothesis pointer fill.

Incorrect
: The instance referred to by the reference pointer fill is not mapped to the instance referred to by the hypothesis pointer fill.

Missing
: the reference pointer fill is mapped to nothing.

Spurious
: the hypothesis pointer fill is mapped to nothing.

(Unscored) Optional
: the reference pointer fill is mapped to nothing, but is in a slot marked optional.

(Unscored) Removed
: the reference pointer fill is mapped to nothing, but the (reference) instance referred to by the reference pointer fill was marked optional, and no hypothesis instance was mapped to the optional reference instance.

Comparison of Text Fills for Content (Clean Data)

There are two ways to compare text fills. One way is by *content*, and one is by *extent*. Mapping one reference text fill to one hypothesis text fill can result in either one or two points. If only contents or only extents are compared, then there is one point per single fill mapping. If both content and extent are compared, then there are two points per mapping.

To compare content, the hypothesis content string is checked to see if

1. it is a substring of the reference maximal content string, and
2. there is a reference minimal content string which is a substring of the hypothesis content string.

If both of these conditions are met, the content point is correct. Otherwise it is incorrect.

Comparison of Text Fills for Content (Noisy Data)

When the reference and hypothesis source documents are recordings of human speech, the corresponding text documents often contain words that should not be taken into account when comparing content. Pause fillers like "uh" and incomplete words like the one in "Glen Buni- Bunting" should be treated as optional content words. If they are in a reference content fill (maximal or minimal) but not in the hypothesis content fill, they should be ignored.

When comparing content for noisy data, the content string will be broken into tokens, some of which may be optional. Then proceed as for clean data, except that strings of tokens will be compared, rather than strings of characters, and some of the reference tokens will be optional.

Comparison of Text Fills for Extent

To compare extents, we proceed as follows. If the reference fill's maximal extents enclose the hypothesis fill's extents, and the hypothesis fill's extents overlap one of the reference minimal fill's extents, the extent point is correct. Otherwise, it is incorrect.

To compare extent, the hypothesis extent is checked to see if:

1. it is enclosed in the reference maximal extent, and
2. it overlaps some reference minimal extent.

If both of these conditions are met, the extent point is correct. Otherwise it is incorrect.

Single Fills May Not Be Split

The content and extent points are determined independently for a single fill pair. However, as stated previously, only one single fill from the reference may map to the single fill in the hypothesis. If the system locates the correct extent, but the contents at that location differ, the content of a different alternative which matches may not be mapped. For example, if two date slots were mapped like this:

```
REFERENCE                         HYPOTHESIS
DATE:  thirsty   ##10#20#         thirsty   ##99#109#
      /thursday  ##99#109#
```

then either the extent or the content could be counted correct, but not both.

Calculation of Scores based on the Mapping

Given a set of points, there are several values calculated in the alignment and final scoring.

`COR` Correct
: The number of correct points

`INC` Incorrect
: The number of incorrect points

`MIS` Missing
: The number of missing points

`SPU` Spurious
: The number of spurious points

`POS` Possible
: The number of points from single fill mappings which contained a reference single fill.

```
POS = COR + INC + MIS
```

`ACT` Actual
: The number of points from single fill mappings which contained a hypothesis single fill.

```
ACT = COR + INC + SPU
```

`REC` Recall
: a measure of how many of the reference fills were produced in the hypothesis.

```
      COR
REC = ---
```

```
       POS
```

PRE Precision

a measure of how many of the hypothesis fills are actually in the reference.

```
PRE =  COR
       ---
       ACT
```

F-measure

a function used to combine recall and precision measures into one measure [VANRIJSBERGEN] . The formula for F is

```
                  2
       ((beta)   +   1.0) * PRE * REC
  F = -----------------------------
                     2
          ((beta)   * PRE) + REC
```

where beta is the relative weight of precision and recall. When precision and recall are given equal weight, the value for beta is 1. Substituting 1 for beta, and the previous formulas for precision and recall, the above formula simplifies to

```
        2 * COR
  F = ---------
       POS + ACT
```

The following measures are also calculated from the points:

UND Undergeneration

```
UND =  MIS
       ---
       POS
```

OVG Overgeneration

```
OVG =  SPU
       ---
       ACT
```

SUB Substitution

```
SUB =  INC
       ---
       COR
```

ERR Error per response fill

```
        INC + SPU + MIS
```

```
     ERR = ---------------------
                 COR + INC + SPU + MIS
```

Sample Text Corpus

Here is a hypothetical text corpus for an imaginary sports event templette task:

```
<DOC>
<DOCNO> ABC19980307.1830.1415 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 03/07/1998 18:53:35.76 </DATE_TIME>
<BODY>
<HEADLINE>
SPORTS
</HEADLINE>
 Byline:JOHN FRANKEL, AARON BROWN
 High:MIKE TYSON SUES DON KING FOR MISMANAGEMENT
 Spec:SPORTS / CASEY MARTIN / BASEBALL / MIKE TYSON
[USE # 4 OF THIS PREAMBLE]
<TEXT>
And one final sports note -- today in
Anchorage,...
<ANNOTATION>  (voice-over)  </ANNOTATION>
...Alaska, the ceremonial start of the 26th
Iditarod dog sled race.  The race officially
starts tomorrow when 63 teams take off on their
1,100 mile trek to Nome.  First price is worth
$50,000.
<ANNOTATION>  (on camera)  </ANNOTATION>
And I think the whole crew and I are happy we are
here in warmer Austin, Texas.  That does it for
sports.
Aaron?
<TURN>
<ANNOTATION> spkr:AARON_BROWN </ANNOTATION>
John, thank you very much.
</TEXT>
</BODY>
<END_TIME> 03/07/1998 18:54:01.67 </END_TIME>
</DOC>
<DOC>
<DOCNO> PRI19980302.2000.2923 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 03/02/1998 20:48:43.85 </DATE_TIME>
<BODY>
<TEXT>
The followup now to a story we reported last
Friday. Egypt has won its first African cup of
nation soccer tournament in 12 years over the
weekend.  Beating defending champion south Africa
2-0. Egypt, which failed to qualify for this
Summer's world cup, also won the African cup title
in 1957, 1959, and 1986.  The host of the
tournament turned in a stellar performance,
placing fourth.
</TEXT>
</BODY>
<END_TIME> 03/02/1998 20:49:16.69 </END_TIME>
```

```
</DOC>
<DOC>
<DOCNO> PRI19980317.2000.2025 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 03/17/1998 20:33:45.13 </DATE_TIME>
<BODY>
<TEXT>
In Boston, I'm Lisa mullens. For a couple of years
now, Boris Yeltsin's health has prompted concerns
about his ability to govern Russia. Just last
Wednesday Yeltsin offered to prove to reporters
that he is in perfect shape.
<TURN>
Tell me the kind of sport you want me to challenge
you in and I'm on my way to the sports
ground. Tell me. Let's go to the swimming pool, to
a tennis court or to a running track. Let's do it.
<TURN>
But today Boris Yeltsin's latest illness forced
the postponement of Thursday's scheduled summit of
the presidents of former Soviet republics. The
Russian leader said to have a severe cold and a
bad cough. The president's health isn't as good as
he'd like people to think.
</TEXT>
</BODY>
<END_TIME> 03/17/1998 20:37:31.82 </END_TIME>
</DOC>
```

Sample Reference Template Set

Here is a sample reference template set, corresponding to the above text corpus.

```
<TEMPLATE-ABC19980307.1830.1415-1> :=
    DOC_NR: ABC19980307.1830.1415 ##14#35#
    COMMENT: "Race hasn't started yet"
<TEMPLATE-PRI19980317.2000.2025-1> :=
    DOC_NR: PRI19980317.2000.2025 ##14#35#
    COMMENT: "Challenges only, no event"
<TEMPLATE-PRI19980302.2000.2923-1> :=
    DOC_NR: PRI19980302.2000.2923 ##14#35#
    EVENT: <SPORTS_EVENT-PRI19980302.2000.2923-1>
    COMMENT: "No locations for earlier tournaments."
<SPORTS_EVENT-PRI19980302.2000.2923-1> :=
    S_EVENT: "African cup of nation soccer tournament" ##216#255#
           / "the African cup" ##401#416#
           / "the tournament" ##461#475#
    WINNER: "Egypt" ##332#337#
    LOSER: "defending champion [south Africa]" ##295#326#314#326#
    SCORE: "2-0" ##327#330#
    LOCATION: "south Africa" ##314#326#
            / "The host of the tournament" ##449#475#
    DATE: "03/02/1998" ##89#99#
    COMMENT: "location of earlier tournaments unstated"
```

Sample Hypothesis Template Set

Here is a sample hypothesis template set, corresponding to the above text corpus.

```
<TEMPLATE-ABC19980307.1830.1415-1> :=
    DOC_NR: ABC19980307.1830.1415 ##14#35#
<TEMPLATE-PRI19980317.2000.2025-1> :=
    DOC_NR: PRI19980317.2000.2025 ##14#35#
<TEMPLATE-PRI19980302.2000.2923-1> :=
    DOC_NR: PRI19980302.2000.2923 ##14#35#
    EVENT: <SPORTS_EVENT-PRI19980302.2000.2923-1>
<SPORTS_EVENT-PRI19980302.2000.2923-1> :=
    S_EVENT: "African cup of nation soccer tournament" ##216#255#
    WINNER: "Egypt" ##332#337#
    LOSER: "defending champion south Africa" ##295#326#
    SCORE: "2-0" ##327#330#
    LOCATION: "south Africa" ##314#326#
    DATE: "03/02/1998" ##89#99#
```

References

BURGER_NAMED
"Named Entity Scoring for Speech Input", John D. Burger, David Palmer, Lynette Hirschman, the MITRE Corporation, 202 Burlington Road, Bedford MA 01730, USA.

CHINCHOR_FOUR
"Four Scorers and Seven Years Ago: The Scoring Method for MUC-6", Nancy Chinchor and Gary Dungca, Proceedings of the Sixth Message Understanding Conference, November 6-8, 1995, Columbia, MD (sponsored by DARPA), Morgan Kaufmann Publishers, San Francisco, ISBN 1-55860-402-2, pp. 33-38.

DOUTHAT_MESSAGE
"The Message Understand Conference Scoring Software User's Manual", Aaron Douthat, ftp://ftp.muc.saic.com/pub/MUC/scorer/MUC_scorer3.3.tar.gz

FISHER_BETTER
"Better Alignment Procedures for Speech Recognition Evaluation", by W. M. Fisher and J.G. Fiscus, IEEE International Conference on Acoustics, Speech, and Signal Processing 1993, pp. II-59 - II-62.

FISHER_FURTHER
"Further Studies in Phonological Scoring", by W. M. Fisher, J. Fiscus, A. Martin, D. S. Pallett, and M. A. Przybocki, Proceedings of the Spoken Language Systems Technology Workshop, January 22-25, 1995, Austin, TX (sponsored by ARPA), Morgan Kaufmann Publishers, San Francisco, ISBN 1-55860-374-3, pp. 181-186.

FISHER_TALD3E
The "tald3e_sm_export" program documentation, by W. M. Fisher. ftp://jaguar.ncsl.nist.gov/pub/aldistsm-1.1.tar.Z

HIRSCHMAN_HUB4
"Hub 4 Event99 General Guidelines", by Lynette Hirschman, Patricia Robinson, Lisa Ferro, Nancy Chinchor, Erica Brown, Ralph Grishman, and Beth Sundheim.

MITRE_MSCORE
The MITRE Named Entity Scorer, http://www.mitre.org/cgi-bin/get_mscore/

NIST_HUB4
"The 1998 Hub-4 Evaluation Plan for Recognition of Broadcast News, in English", http://www.nist.gov/speech/hub4_98/hub4e_98_spec.htm

PICONE_AUTO
"Automatic Text Alignment for Speech System Evaluation", Joseph Picone, Kathleen M.

Goudie-Marshall, George R. Doddington, and William Fisher, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 4, August 1986.

VANRIJSBERGEN

Van Rijsbergen, C. J. (1979) Information Retrieval. London: Butterworths.

Author

Aaron L. Douthat, SAIC (AARON.L.DOUTHAT@cpmx.saic.com)

Last modified: Thu Feb 18 14:34:48 1999