Do Scripts solve NLP?

A Discussion on the Use of Scripts to Understand the MUC3 Kidnapping Articles

Peter Clark, Boeing Phantom Works, Seattle, WA
peter.e.clark@boeing.com
June 2008

1. Introduction

The performance task used for the Third Message Understanding Conference (MUC-3) in 1991 (MUC3, 1991; Lehnert and Sundheim, 1991) was information extraction from newswire articles about terrorism. Given the constrained domain, it seems like just matching a script or two against the text should work well. This note explores that hypothesis, and also looks briefly at how MUC-3 participants actually addressed the challenge.

Let us narrow the task even further, and consider just stories about kidnappings and the single question "Who were the perpetrators?" (one of the template questions used in MUC-3). A simple (semi-formal) script for kidnapping is shown in Appendix 1. Scripts can work quite nicely: Consider the following (hand-constructed) RTE-style example:

> **T:** Fernando Lopez was kidnapped Tuesday. The LFA sent a message to a friend of Lopez, demanding one million dollars ransom.
> **H:** Lopez was kidnapped by the LFA.

Using an even more simplified script (abstracted from Appendix 1), our RTE system can answer "Yes" to the above example, by reasoning that:

> (i) There's (probably) a kidnapping going on (from the word "kidnapped"), so the kidnapping script applies
> (ii) This creates the expectation (from the script) that the kidnapper will send a message
> (iii) T says the LFA sent a message
> (iv) Therefore LFA must be the kidnapper.

The simplified script used in this example looks:

> There is a kidnapping.
> The kidnapper makes a plan for a kidnapping.
> The kidnapper approaches the victim.
> The kidnapper captures the victim.
> The kidnapper takes the victim to a hideout.
> The kidnapper sends a message to an acquaintance of the victim.
> The message contains a demand and a threat.
> The acquaintance reads the message.
> The demand is met by the acquaintance.
> The kidnapper releases the victim.
> The kidnapper hides from the police.

The interesting question is can we solve real examples in a similar style?

2. Some MUC3 Examples and Discussion

Here are some MUC-3 kidnapping examples; we discuss how easy/difficult it is to identify the perpetrators of the kidnapping from the texts. The texts are slightly simplified to keep them short while preserving the phenomena we wish to discuss. The answer (to the question "Who were the perpetrators?") is shown in bold. Coreference resolution is needed throughout. We will assume here that it can be done correctly, and similarly assume parsing is done correctly (two rather drastic assumptions, but okay for this thought experiment).

TST1-MUC3-0080

> Liberal Senator Federico Velez was kidnapped on 3 April at the corner of 60th and 48th Streets in Western Medellin. The party leader had left his house only minutes earlier. As he waited for the traffic light to change, **three heavily armed men** forced him to get out of his car and get into a blue Renault.

Discussion:

This requires realizing that "men forcing Velez to get out of his car" is part of "the kidnapper captures the victim" step in the kidnapping script. To do this, we need a whole new script about capturing people, which serves as a "sub-script" of this script. Here it might need to be even more specific, e.g., capturing people in a vehicle, with steps something like:

<u>“Capturing person in vehicle” script</u>
(a) stop the vehicle
(b) remove the occupant from the vehicle

This still doesn't quite match here though:

1. In this example, the kidnappers didn't stop the vehicle, it stopped itself (at a traffic light). In addition, the text doesn't say the vehicle stopped, it says the driver "waited for the traffic light to change". There's some commonsense knowledge needed here that this means that the driver's vehicle must therefore be stopped!

2. How do we match "remove the occupant from the car" (in the hypothetical sub-script) with "three men forced Velez to get out of his car"? Again there's some world knowledge that

   "X forces Y to get out of Z" → "X removes Y from Z"

   (The DIRT paraphase database has no rules for "X forces Y", unfortunately). There may be some generalization of this: Let's say "get out" is a reflexive action, i.e, (Y gets out of Z) → (Y moves Y out of Z). (UT's Component Library explicitly labels some events as being reflexive in this way; they say a "reflexive cliche" applies). Now we can define force as:

   IF X forces Y to do: Y does Action
   THEN we can say: X does Action

   IF X forces Y to do: Y moves Y out of Z *; e.g., men force V to move V from C*
   THEN X moves Y out of Z *; e.g., men moves V from C*

   and "X moves Y out of Z" should then be recognizable as a "remove" event. (Phew!).

DEV-MUC3-0008 (NOSC)

> Castellar was kidnapped on 5 January on the outskirts of Achi, about 850 km North of Bogota, by a group of armed men....Castellar was kidnapped by **ELN guerillas** while he was traveling in a boat down the Cauca River to the Tenche area, a region within his jurisdiction.

Discussion

This is straightforward, the answer is directly in the text.

DEV-MUC3-0025 (NOSC)

> The MNR reports the disappearance and kidnapping of MNR Assistant Secretary General Hector Oqueli Colindres in Guatemala City today, 12 January. According to reports, Oqueli and his companion Gilda Flores, a member of the Social Democratic Party of Guatemala, were abducted by **heavily armed men** between 0630 and 0700 as they were heading to La Aurora airport to board their flight.

Discussion

This is straightforward, needing to realize "abduct" and "kidnap" are (approximately) synonyms. (WordNet indeed includes these as synonyms).

DEV-MUC3-0026 (NOSC)

> The Salvadoran government today deplored the disappearance of Social Democratic leader Hector Oqueli Colindres this morning in Guatemala. According to sources from his party, Hector Oqueli Colindres Secretary of the National Revolutionary Movement (MNR) disappeared today in Guatemala when the vehicle in which he was traveling was intercepted by **heavily armed men** wearing civilian clothing.

Discussion

This is an even more challenging example of finding something in the text which matches "the kidnapper captures the victim" in the script. In this case, the match is very indirect, based on the phrase “men intercepted the vehicle”: intercepting a vehicle doesn't necessarily imply capturing the vehicle's occupants, although we're told here the occupant "disappeared" (in this context meaning was captured; WordNet doesn't contain this meaning).

To solve this problem, it looks again (as in example 0080 earlier) like we need a sub-script describing capturing someone in a vehicle. Intuitively it might be similar to the earlier example:

> <u>“Capturing person in vehicle” script</u>
> (a) stop the vehicle
> (b) remove the occupant from the vehicle

However, we still have a "script matching" challenge: how do we link "intercept the vehicle" with "stop the vehicle"? (In fact, WordNet has "intercept" and "stop" as synonyms, so we might be okay here).

DEV-MUC3-0033 (NOSC)

> The Salvadoran opposition National Revolutionary Movement (MNR) today demanded that Guatemalan President Vinicio Cerezo "thoroughly" investigate the assassination of MNR leader Hector Oqueli and of Guatemalan Attorney Gilda Flores. During a news conference, the MNR Executive Committee condemned the "abominable" crime perpetrated on 12 January in Guatemala by **armed men wearing civilian clothes**. Oqueli, 45, and Gilda Flores, a member of the Democratic Socialist Party (PSD) of Guatemala, were kidnapped while they were on their way to La Aurora Airport in Guatemala City.

Discussion

Here the system would need to realize that "crime" refers to "kidnap" (or in this case "assassination"). Some world knowledge that kidnapping is a crime is needed ("kidnap" isn't a subclass of "crime" in WordNet).

DEV-MUC3-0040 (NOSC)

> Unidentified persons kidnapped 31 people in the strife-torn banana-growing region of Uraba. A **group of heavily armed men** forced the kidnapped villagers to climb into trucks and then took off for an unknown destination.

Discussion

We will focus on the task of identifying the “group of heavily armed men” as kidnappers. (In addition, there is an easy reference to “Unidentified persons” as kidnappers, which we will assume is straightforward to handle).

The original "kidnapping" script included:

> "The kidnapper captures the victim."
> "The kidnapper takes the victim to a hideout."

In this story, the key phrases are:

> "Heavily armed men forced the villagers to climb into trucks."
> "[The men] took off for an unknown destination."

Matching these two looks very difficult, in particular realizing that "climb into trucks" is a substep of "take to a hideout". For a deep understanding, we would need a "transport by vehicle" script, e.g.,:

> <u>“Transport by vehicle” script</u>
> (a) put cargo into vehicle
> (b) get into vehicle
> (c) start engine
> (d) travel (e.g., drive)
> (e) stop
> (f) get out
> (g) unload cargo

In this story, step (a) is achieved somewhat atypically (as the "cargo" is people): the kidnappers don't pick up and place the villagers in the truck, but instead force the villagers themselves to climb into the truck. We could recognize the equivalence using a rule similar to example 0080 earlier:

> **IF** X forces Y to do: Y moves Y into Z
> **THEN** X moves Y into Z

Thus we can recognize that "the heavily armed men forced the villagers to climb into the truck" instantiates step (a) "X puts (moves) cargo into vehicle" (assuming "move into" and "put into" are synonyms), and thus the kidnapper X = "the heavily armed men".

DEV-MUC3-0125 (BELLCORE)

> The report indicates Angel Marroquin Hernandez and Jose Angel Flores, members of the Santo Domingo Canton Civil Defense, were abducted and subsequently murdered by the **rebels** near a bridge in the Guazapa jurisdiction.

Discussion

Straightforward, given "abduct" and "kidnap" are synonyms (true in WordNet).

DEV-MUC3-0183 (ADS)

> Three officials of BrasPetro have been kidnapped by a **group of guerillas** from the Medio Magdalena Petroleum basin, in Colombia.

Discussion

Straightforward.

3. Analysis

1. Despite the above cases where things are not straightforward, it seems that some expectation-based approach is essential for understanding these kinds of stories. Repeating the example at the start of this note:

    > **T:** Fernando Lopez was kidnapped Tuesday. The LFA sent a message to a friend of Lopez, demanding one million dollars ransom.
    > **H:** Lopez was kidnapped by the LFA.

    it is clear some domain knowledge is needed. We suggested this might be a script, encoding what goes on in a kidnapping including that kidnappers demand ransoms. One could encode the script as a sequence of events, but in practice MUC systems typically did something else, just encoding "extraction patterns" such as

    > "*X* demands ransom" → *X* is the kidnapper
    > "*X* asks for ransom" → *X* is the kidnapper
    > etc.

    One could view this as, rather than encoding the script per se, encoding all the surface level manifestations of each script element (which could be – and was – a laborious task; it might be that modern paraphrase technologies could ease this burden). The identification, rationale, order, and coherence of the underlying events was typically not represented, as it was not needed for the specific task (just the surface manifestation was needed).

Sometimes extraction patterns also included type information, e.g., consider "*X* captured *Y*" → *X* is the kidnapper and Y the victim. More accuracy can be obtained by restricting the identities/types of *X* and *Y* (e.g., *X* shouldn't be a terrorist, *Y* should be a terrorist or organization, except not a law enforcement organization (e.g., police).

In MUC-3, patterns were usually authored by hand, but could – to an extent - also be authored using semi-automated techniques. For example, Riloff's AutoSlot-TS system (Riloff 1999) learned patterns by

(i) given a noun phrase in a sentence, automatically enumerate the extraction patterns for it (typically 2 or 3). E.g. "terrorists" in "Terrorists attacked the US embassy" → "*NP* attacked" and "*NP* attacked embassy". The number of possible extraction patterns are small and didn't use type information, but in principle could have done so.
(ii) Repeat for all noun phrases in the terrorism corpus
(iii) See how often each pattern fires in the terrorism corpus vs. a general corpus. "Relevant" patterns are those which fire unusually frequently in the terrorism corpus.
(iv) Rank the patterns based on relevance (proportion of firings) and frequency (number of firings) on the terrorism corpus. Domain-specific patterns should thus end up high, e.g., "*NP* exploded"
(v) For the top patterns, manually label the roles e.g., "*NP* exploded" → *NP* = the instrument.

Essentially the algorithm automatically discovers that, for example, there appear to be lots of things that explode in the terrorism corpus, or that lots of things get kidnapped. The user then labels what those things are.

2. Even in this restricted domain, it is apparent that a single script is not sufficient. Each step in the script can potentially be carried out in many ways; if the story delves into details of that step, then we need to also represent the scripts for those details. In the above examples, for instance, we also need scripts for "capturing people in a vehicle" (to understand the relevance of "forced Velez to get out of his car" and "intercept the vehicle") and "transport by vehicle" (to understand the relevance of "climb into trucks").

3. Even if the script contains a step mentioned in the text, it still can be difficult to match the two, and may require some additional general world knowledge. For example, above, matching "remove the occupant from the vehicle" with "force the occupant to get out of his car" is a non-trivial matching task, and requires either "deep" general knowledge about "force" and "control", or a bunch of surface-level paraphrase rules. As a point of interest, the DIRT paraphrase database does include paraphrases for "remove X from Y" (e.g., "pull X from Y") but "force X to get out of Y" is not one of them. One can view this matching problem as an example of the "recognizing textual entailment" challenge.

4. In contrast to the deep analysis so far, it's worth noting that "cheap tricks" would make some progress. If the story is known to be about kidnapping, then any mention of "armed men", "guerillas", etc., is likely to refer to the kidnappers, even if nothing else is understood about the story. The "kidnapping script" expects a set of players (victim, perpetrators, etc.) and given a story we can go a certain way to identify those players simply based on their descriptions (e.g., victims are often people of power/wealth/political stature; perpetrators are often armed). Or one could go a little further: we expect certain behaviors by the players (e.g., victims are often scared, perpetrators often attack), and so could do a partial linguistic

analysis to identify the players based on what they do (i.e., the subject-verb pairs in the text) without having to fully parse the sentences. In fact, Schank advocated this kind of "skimming" in his work on scripts, resulting in systems like FRUMP (DeJong 1979) and NOMAD (Granger, 1982).

In fact, it is somewhat unfair to call these word/phrase-spotting approaches "cheap tricks"; really they are just lighter-weight ways of matching a script with text.

One could imagine combining "lightweight" and "heavyweight" approaches to matching scripts with text. A lightweight approach might see "armed men" and guess they were the perpetrators (using no other evidence); a heavyweight approach might match a sentence mentioning armed men with part of the kidnapping script, again identifying the armed men as perpetrators. Both approaches provide evidence, and hence we may get a more robust system by combining this evidence together rather than being just "heavyweight" (full parse) alone. In particular, when sentences cannot be parsed, or are misparsed, or alternative parses are possible, use of additional evidence like this could potentially be very useful.

5. Lightweight methods could also be used for *learning* scripts. Rather than parsing a sentence, then seeing if the parse matches an expected relationship between the perpetrator and victim (say), and thus identifying the perpetrator and victim, one could do the reverse: First guess the perpetrator and victim based on surface attributes, then *find* relationships between the two in the article (thus the system is *learning* perpetrator-victim relationships). Or the two directions could be dovetailed:

    a. match an observed relationship with an expected relationship to find the perpetrator and victim (say)
    b. collect *other* relationships between the perpetrator and victim in the text, to augment the set of expectations in the script. These other relationships might be alternative ways (paraphrases) of expressing a known relationship in the script, or may be a new relationship not mentioned in the script. (Some user interaction might be needed to distinguish between the two).

   A similar approach has been very effectively used for learning how a known relationship (e.g., date of birth) manifests itself in language (e.g., Snow, Jurafsky, and Ng, 2005; Snow, Jurafsky, and Ng, 2006). First, seed a system with known example(s) of that relationship (e.g., Mozart-27Jan1756), either entered by hand, or found by looking for a pattern unambiguously expressing this relationship (e.g., "*person* was born in *date*"). Then look for other patterns/relationships involving these pair(s), (e.g., "*person* (*date-date2*)..."). Conjecturally, the high frequency patterns are also likely to express that relationship. One can view this as a special case of paraphrase learning (Lin and Pantel, 2001), focusing on a specific relationship to learn.

6. It is worth noting the high linguistic complexity of the language in these examples (i.e., real language). In the above analysis I've assumed a complete and perfect parse, and perfect intersentence coreference resolution, but in practice this is very difficult and expensive to achieve. Even in this relatively restricted domain, it took the participating teams 6 months of intensive system engineering to get reasonable performance. Riloff (1999) writes that “even the most linguistically-oriented Information Extraction researchers have reluctantly accepted that, for the information extraction task at least, the potential benefits of full parsing are usually overwhelmed by the extra overhead and lack of robustness.” Given this challenge, it seems even more important to use shallow as well as "deep" approaches to understand the

text. In fact, the MUC3 systems were reported to cover a broad spectrum of approaches from shallow analysis to full semantic parsing (Lehnert and Sundheim, 1991).

In addition to general parsing, this particular task requires handling some more specialized language, specifically: dates, names, titles of political positions, locations, and newswire-specific formats (e.g., how the article's author, origin, and date are presented). Task-specific lexical information would also significantly help (e.g., people, organizations, and place names relevant to the corpus, namely Latin American terrorism from May 1988 to May 1990). In addition (reflecting the age of MUC3) the newswire articles were all upper case, making parsing even harder. (The quoted articles above have had case-sensitivity added for readability in this document).

7. Although MUC3 (presumably) stipulated that each story be processed in isolation, it is clear from the above examples that there is redundancy in the entire collection, as several articles report the same event. For example, articles 0025, 0026, and 0033 above all describe the same event (kidnapping and murder of Hector Aqueli Colindres). This redundancy could be exploited to improve robustness, e.g., to overcome failures and errors on specific sentences, or to resolve ambiguity. For example, in our analysis we pointed out that is would be very challenging for a system to realize that the "armed men" were the perpetrators in 0026 and 0033, but in 0025 it is easy as it is stated directly ("Oqueli was abducted by **heavily armed men**"). One could thus use knowledge from one article to help disambiguate processing of another article, and maybe also allow knowledge from that article to feed back into the knowledge base of scripts/expectations.

3. Summary

Despite its age, MUC3 (and the other MUCs) is an important and informative datapoint about the capabilities of, and challenges for, NLP systems. In general, it appears that the MUC3 systems were heavily engineered to work with the very specific style and topic of the articles, and so had limited potential for direct transfer and generality, although clearly the general methods employed could be (and many were) reused (with substantial reengineering work) for new tasks. (Some progress was also made on reducing the manual engineering effort required for new domains, such as AutoSlog-TS described earlier). Even then, the systems' scores were far from perfect: According to Lehnert and Sundheim (1991), recall averaged 26% and precision 52% for the MUC3 systems, with only two systems achieving recall over 40% and precision over 60%; thus even the best systems were getting an answer wrong approximately as often as getting an answer right. While these scores are impressive from a technical point of view, they also reflect the vast challenges of processing language. As a caveat, MUC3 is now over 15 years old; it is likely that modern systems would perform substantially better if MUC3 was rerun today.

The degree of domain- and task-specific engineering required for MUC3 is somewhat daunting, and runs counter to the dream of building simple, domain-general language understanders. It appears that a substantial part of MUC-3 involved frantically encoding all the possible surface manifestations of (i.e., extraction patterns for) entities and relations of interest. While clearly there has to be substantial domain knowledge injected somewhere, other approaches might have be possible, e.g., learning extraction patterns automatically; deriving them from a deeper representation of a script; exploiting paraphrase databases and core theories to identify equivalences between the text and the expected surface pattern/underlying event.

Also, as we have illustrated, encoding "deeper" representations of the scripts itself not a simple task. In particular, there is not a single "kidnapping" script, but rather any specific kidnapping is a composition of a general kidnapping script with other sub-scripts showing how details are carried out – essentially the same observation that Schank arrived at when his ideas on Scripts evolved to MOPs (Memory Organization Packets) (Schank, 1992).

To summarize the main points of the discussion:

1. MUC-3 systems typically didn't encode deep knowledge (scripts) per se, but just extraction patterns for the many surface-level realizations of script elements.
2. Matching a single script against text is a somewhat naïve model; rather each specific kidnapping story is a (typically novel) composition of several scripts.
3. Matching a particular scriptal expectation against text is itself challenging, any may require use of world knowledge. This task is similar to the "recognizing textual entailment" challenge.
4. "Cheap tricks" exploiting surface features (e.g., "armed men" → kidnapper) shouldn't be thrown out, they can help and in fact are really just another aspect of expectation-driven understanding.
5. Scripts and extraction patterns can be learned, as well as encoded.
6. Simply parsing the text (which we have not focused on here) is itself challenging; MUC-3 systems were typically pipelined, thus requiring correct parses up front, and another substantial part of the development effort was simply getting the basic linguistic processing for this target domain to work adequately. In the long term, a simple pipeline model is unlikely to be adequate (unless there is redundancy so nonsensical parses can be rejected).
7. MUC-3 did nothing to exploit redundancy among multiple texts; this could be an important aspect of a larger machine reading program.

References

G. DeJong "Prediction and Substantiation: A New Approach to Natural Language Processing" Cognitive Science 3(3) July 1979 , pp251-271.

E. Domeshek, E. Jones, and A. Ram. (1999) "Capturing the Contents of Complex Narratives", in Understanding Language Understanding, p73-106, Ed. A. Ram and K. Moorman. MA: MIT Press.

R. H. Granger "Scruffy Text Understanding: Design and Implementation of 'Tolerant' Understanders." Proc 20th ACL (1982) pp157-160.

W. Lehnert and B. Sundheim. "Performance Evaluation of Text-Analysis Technologies" AI Magazine 12 (3) (1991) pp81-94.

D. Lin and P. Pantel. "Discovery of Inference Rules for Question Answering", Natural Language Engineering 7(4), pp343-360 (2001).

E. Riloff. (1999) "Information Extraction as a Stepping Stone toward Story Understanding", in Understanding Language Understanding, Ed. A. Ram and K. Moorman. MA: MIT Press.

R. Schank (1982) “Dynamic Memory”.

R. Snow, D. Jurafsky, and A. Y. Ng. 2005. "Learning syntactic patterns for automatic hypernym discovery", in NIPS 2005.

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. "Semantic taxonomy induction from heterogenous evidence". Proceedings of COLING/ACL 2006, Sydney.

MUC3 (1991). The MUC-3 dataset is available at: http://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html

Appendix 1: A richer script for kidnapping

SCRIPT-TITLE:
A kidnapper kidnaps a victim.

PARTICIPANTS:
A kidnapper,
A victim,
An acquaintance of the victim.

ASSUME:
The kidnapper is an adult human or a group of adult humans.
The victim is a human or a group of humans. [any age]
The acquaintance is an adult human or a group of adult humans.

BEFORE-THE-SCRIPT:
The kidnapper does not control the victim.

GOAL: [of kidnapper]
The kidnapper receives money from someone who knows the victim.
OR
The kidnapper extracts the release of prisoners in exchange for the victim.
OR
The kidnapper extracts information from the victim.
OR
The kidnapper sexually attacks the victim.
OR
The kidnapper adopts the victim as his/her own child.
OR
The kidnapper murders the victim in revenge.

GOAL: [of victim]
The victim is released unharmed by the kidnapper.

BEGIN-SCRIPT:
The kidnapper makes a plan for a kidnapping.
The kidnapper approaches the victim.
The kidnapper gains control of the victim.
The kidnapper takes the victim to a hideout.
{ The kidnapper sends a message to an acquaintance of the victim.
The message contains a demand and a threat.
The acquaintance reads the message.
The acquaintance decides whether to meet the demand.
{ The demand is met by the acquaintance.
The kidnapper releases the victim.
OR The kidnapper kills the victim.
OR The kidnapper injures the victim. }
OR
{ The demand is not met by the acquaintance.

The kidnapper kills the victim.
OR The kidnapper injures the victim. }
OR
{ The kidnapper demands information from the victim.
The kidnapper threatens to torture the victim.
The victim gives the information to the kidnapper.
OR { REPEAT The kidnapper tortures the victim.
UNTIL The victim gives the information to the kidnapper. }
The kidnapper releases the victim.
OR The kidnapper kills the victim.
OR The kidnapper injures the victim. }
OR
The kidnapper sexually attacks the victim.
The kidnapper releases the victim.
OR The kidnapper kills the victim.
OR The kidnapper injures the victim.
OR
The kidnapper raises the victim as his/her own child.
OR
The kidnapper kills the victim in revenge.
The kidnapper hides from the police.

END-SCRIPT

AFTER-THE-SCRIPT:
[too many different endings to say anything here.]