School of Computing and Information Systems
The University of Melbourne
COMP90042 WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2019)

Workshop sample solutions: Week 8

Discussion

1. What is **Information Extraction**? What might the "extracted" information look like?

   - Basically, we want to extract information from a (generally unstructured) document, into a structured format that we can sensibly query later.

   (a) What is **Named Entity Recognition** and why is it difficult? What might make it more difficult for persons rather than places, and *vice versa*?

   - We want to find **named entities** — mostly proper nouns, but also sometimes times or numerical values of significance — within a document. Often context (which is not always present within the sentence, or even the document!) is needed to disambiguate the type of named entity, or even whether a given phrase is a named entity at all.
   - One common problem, that we see with both people's names and places, is that they are ambiguous with common nouns. Generally speaking, we can write a (somewhat) exhaustive list of names of places — a **gazetteer** — but we can't with names of people, which are constantly changing. On the other hand, many different locations can have the same name (e.g. Melbourne, Australia and Melbourne, USA), whereas this tends to happen to a lesser extent for people's names, especially in formal text (like in a newspaper).

   (b) What is the **IOB** trick, in a sequence labelling context? Why is it important?

   - Named entities often comprise multiple tokens, like "the University of Melbourne" — these are often represented through bracketing, e.g. `I visited [the University of Melbourne]`$_{\text{LOC}}$ `yesterday.`
   - Getting the bracketed entity from a tree structure — like one produced from **parsing** — is fairly straightword, but it is often more convenient to **tag** individual tokens. To do this, we indicate whether a given token is **B**eginning a named entity, **I**nside a named entity, or **O**utside a named entity, so that the sentence above might look like `I-O visited-O the-B-LOC University-I-LOC of-I-LOC Melbourne-I-LOC yesterday-O .-O`

   (c) What is **Relation Extraction**? How is it similar to NER, and how is it different?

   - Relation Extraction attempts to find and list the relationships between important events or entities within a document.
   - Relations typically hold between entities (e.g., MP-for(Turnbull, Wentworth)), so in order to extract relations you first need to do NER to extract the entities from the text (e.g., **Turnbull**, the member of **Wentworth**, said ...; where the bolded items would be tagged using an NER system.)

(d) Why are hand–written patterns generally inadequate for IE, and what other approaches can we take?

- Basically, because there are too many different ways of expressing the same information. We can often write rules that lead to accurate relations (high Precision), but they won't cover all of the relations within an unseen document (low Recall).
- Parsing the sentence might lead a more systematic method of deriving all of the relations, but language variations mean that it's still quite difficult. More sophisticated approaches frame the problem as supervised machine learning, with general features (like POS tags, NE tags, etc.) and features specific to the relations that we're trying to identify.
- **Bootstrapping** patterns — using known relations to derive sentence structures that describe the relationship — is also somewhat popular in some domains.

2. What is **Question Answering**, and how is it related to **Information Retrieval** and Information Extraction?

- In short, we want to use our **knowledge base** — either in terms of raw documents, or in relations that we've already extracted from the documents — to answer questions (perhaps implicitly) posed by a user.

(a) What is **semantic parsing**, and why might it be desirable for QA? Why might approaches like NER be more desirable?

- As opposed to **syntactic parsing** — which attempts to define the structural relationship between elements of a sentence — we instead want to define the (meaning–based) relations between those elements.
- For example, in the sentence `Donald Trump is president of the United States.` we can deduce that `Donald Trump` is the subject of the verb `is`, and so on, but in semantic parsing, we might be trying to generate a logical relationship like `is(Donald Trump, president(United States))`.
- This format allows us to answer questions like "Who is president of the United States?" by generating an equivalent representation like: `is(?,president(United States))`

(b) What might be the main steps for answering a question for a QA system?

- In a Relation Extraction sense:
  - Offline, we process our document collection to generate a list of relations (our knowledge base)
  - When we receive a (textual) query, we transform it into the same structural representation, with some known field(s) and some missing field(s)
  - We examine our knowledge base for facts that match the known fields
  - We rephrase the query as an answer with the missing field(s) filled in from the matching facts from the knowledge base
- In an Information Retrieval sense:
  - Offline, we process our document collection into a suitable format for IR querying (e.g. inverted index)

- When we receive a (textual) query, we remove irrelevant terms, and (possibly) expand the query with related terms
- We select the best document(s) from the collection based on our querying model (e.g. TF-IDF with cosine similarity)
- We identify one or more snippets from the best document(s) that match the query terms, to form an answer