School of Computing and Information Systems
The University of Melbourne
COMP90042 NATURAL LANGUAGE PROCESSING (Semester 1, 2020)

Sample solutions: Week 11

Discussion

1. What is **Question Answering**?

   - QA is the task of using knowledge — 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 are the main steps for answering a question for a QA system?

   - Knowledge-based QA:
     - 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
   - IR-based QA:
     - 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

2. What is a **Topic Model**?

   - A topic model is an unsupervised model that discovers latent thematic structure in document collections.

   (a) What is the **Latent Dirichlet Allocation**, and what are its strengths?

   - LDA is a particular implementation of topic model. LDA is a probabilistic model that assumes each document has a mixture of topics (in the form of a probability distribution), and each topic has a mixture of words (also a probability distribution). Due to its Bayesian formulation (by giving priors to the two aforementioned distributions), LDA is able to infer topics for unseen documents, a capability that its predecessors do not have.

   (b) What are the different approaches to evaluating a topic model?

   - As topic models are unsupervised models, there is no task-based metrics such as accuracy to evaluate them. The best way is to look at the performance of downstream tasks or applications of interest (extrinsic evaluation).
   - Other intrinsic evaluation approaches include:
     - Perplexity: a normalised model logprobability metric over test data.
     - Topic coherence: assess how coherent or interpretable the extracted topics are. We can do this manually with word intrusion (injecting random a word into topic and try to guess which is the injected word) or automatically with PMI measures.

Programming

1. In the iPython notebook `12-topic-model`, we build a topic model on the Reuters news corpus.

   - Explore different number of topics: qualitatively how does it change the topics?
   - Explore different values of the document-topic α and topic-word η (β in lecture) priors: qualitatively how does it change the topics? What values work best for the downstream document classification task? (Note: you can also try 'auto' where the model will try to learn these hyper-parameters automatically)
   - Modify the classification task such that it uses bag-of-word *and* the topic distribution as input features to the classifiers. Do you see a performance gain?