CSE 538 Midterm-2
Fall 2021.

Points This exam is for 100 points.

1. This is an open ended exam. Write as much or as little as you want. I'd expect the whole exam to take somewhere between 8-12 pages total but that is a very rough estimation on my part! You won't be penalized for turning in a shorter document. You cannot exceed 16 pages total.
2. This is a test of whether you are able to apply concepts you have learned in class in some problem settings. For every answer, make sure you first convey your ideas at a high-level clearly. The more the details, the more convincing your answer will be. Draw figures where possible and clearly indicate the input/output to each component in your figure.
3. Questions are intentionally a bit underspecified. You can make assumptions as you need. In some cases I specifically mention some things that you cannot assume.
4. Take home only means that you can do this at your leisure but the solution still **must be yours and yours only**.
5. You **cannot** discuss ideas with your friends. Note it is very easy to find out if you discussed ideas.
6. Questions must not be shared publicly or privately with others.
7. You can use material we discussed in class. Lecture slides, notes and ideas contained in them can be reused but you should use your own articulation of these.
8. You can use ideas from existing papers but you will have to cite those papers. If you use ideas from any paper or lecture notes, mention it right alongside your answer.
9. Please turn in a typed assignment as a pdf file. You can use hand-written equations or figures in the pdf (scanned in) but please make sure they are legible in the final pdf.
10. You can use the latex template of this pdf (also released on Blackboard) for your submission.
11. **The solutions are due 5 pm November 23rd (Tuesday).**

12. You have 24 hours. You can submit until 5pm Nov 23rd without any penalty. If for whatever reason your submission isn't in by 5pm and is delayed we will deduct 20 points for submissions that are made within 6pm – even if it is delayed by only one minute. Submissions made after 6pm won't be graded. You will get a zero.

    It is up to you to ensure that you submit early enough that you don't get caught in last minute blackboard or other issues.

13. **What if I have questions?** Please post your questions here. Do not ask leading questions. You should only ask clarification questions.

    We will respond to questions 9pm Nov 22nd.

    `https://goo.gl/RtYieZ`

    Do not email questions or post on Piazza.

1 Summarization System for Shaltanacs (40 points)

Congratulations! You have been hired by the Shaltanacs for building an abstractive summarization system. The input is a story and the output is always a single sentence describing the story.

The Shaltanian language is similar to English in terms of syntax and the order in which words are laid out. For example, here is a Shaltanian sentence:
`Ito flonn bvoozle grentlyde vroond grizny.`

The corresponding English sentence is:
`That green apple rotted very quickly.`

But Shaltanian's have a huge vocabulary. This means that the decoder in the abstractive summarization system is a huge bottleneck. Where each decoding step i.e., the time it takes to score the entire vocabulary, takes as much as 5 seconds.

Because of quirks in their reading preferences and the time it takes for them to decode, the Shaltanacs want the words to appear in an order that conveys the main contents of a sentence first and then followed by the other words i.e. they want head words of constituent phrases first followed by the modifiers. Note they want the layout of the sentence to be the same i.e. each word should appear in the same position in the sentence but just that the order in which they appear to be different.

So if the above Shaltanian sentence were to be the output summary sentence then it would be generated as follows. First step is to have empty slots that correspond to the number of words that are to be generated. Then, in step 2 the decoder would generate the head word of the entire sentence, the main verb grentlyde in this case. In step 3 it generates the main subject `bvoozle` and then the object `grizny`. Basically, it does a breadth-first expansion based on the syntactic structure of the sentence.

1. T1: `--- --- --- --- --- ---`
2. T2: `--- --- --- grentlyde --- ---`
3. T3: `--- --- bvoozle grentlyde --- ---`
4. T4: `--- --- bvoozle grentlyde --- grizny`
5. T5: `--- flonn bvoozle grentlyde --- grizny`
6. T6: `Ito flonn bvoozle grentlyde vroond grizny`

If this procedure were to be used for the words in the corresponding English sentence: `That green apple rotted very quickly.`, then we will see the words appearing in the following order.

1. T1: `--- --- --- --- --- ---`
2. T2: `--- --- --- rotted --- ---`
3. T3: `--- --- apple rotted --- ---`
4. T4: `--- --- apple rotted --- quickly`

5. T5: `--- green apple rotted --- quickly`
6. T6: `That green apple rotted very quickly`

Note that your decoder can't wait to generate the whole sentence first and then re-order in this fashion. So your system has to be set up so that the decoder generates words in this order at each time step.

Resources Available to you:

1. The Shaltanian Story Summarization corpus consists of 30,000 stories, each paired with a single sentence summary.
2. Apart from the story/summary pair training corpus you don't have any other Shaltanian resource but you can assume you have access any NLP tool you want in English including a English to Shaltanian translator.
3. But you DON'T have any resources in Shaltanian other than the Story Summarization corpus.
4. You also DON'T have a Shaltanian to English translator and can't use that as part of your solution.

Your task:

1. Provide a design of the full system including the encoder and decoder parts for this summarization system. Please provide both a figure that clearly identifies the main components and how they connect with each other. Mark the inputs and outputs clearly.
2. Please identify the design choices you have in building this system and motivate that specific choices you have made.
3. How will you train this system? Describe the training data you will use and how you will obtain it. Please describe in detail how this will be done.
4. Respect the constraints provided in this problem but you can make assumptions where the problem is under-specified. You must state why your assumptions are reasonable.
5. What are the key challenges you will face in developing this system? What are going to be the limitations of your system? (10 points)

Hints:

1. Think about what information your decoder needs to have in order to generate the outputs as required in the problem statement.
2. Think about how you can get this information given the constraints you have on the Shaltanian language resources and what you can transfer from English and how you can do it.

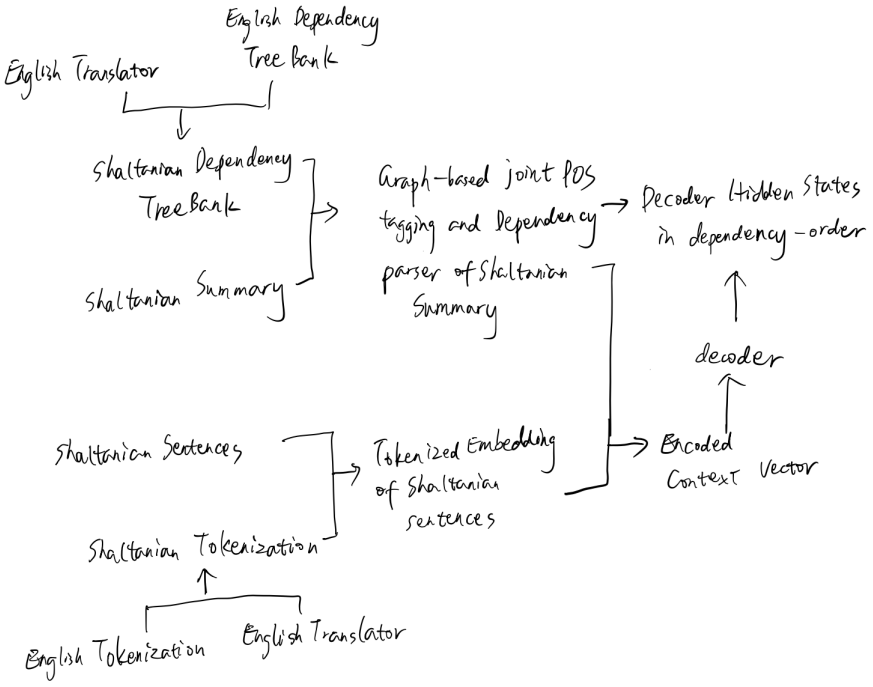

Figure 1: Summarization System for Shaltanacs

Answers:
See Figure 1 for the System of Shaltanacs Summarization:

1. **Motivation**
   For ordered output, we need their dependency relationship of the output, thus I firstly translate the English Dependency Tree Bank into Shaltanian. Then we can generate a Graph-based joint POS tagging and Dependency Parser[Nguyen et al., 2017] for the summaries.
   In order to tokenize the Shaltanian sentences, we also translate English tokens and use their mechanism to do the tokenization. So that we can get the tokenized embedding of Shaltanian sentences.
   For better captures the contextual meaning, here we will also transform the tokenized embedding to a single-layer bidirectional LSTM.

2. **Training**
   Here we use a Sequence-to-sequence attentional model and a pointer-generator network[See et al., 2017] for training. See Figure 2 for the model construction given in the paper. But we would use different way to encode the summaries as

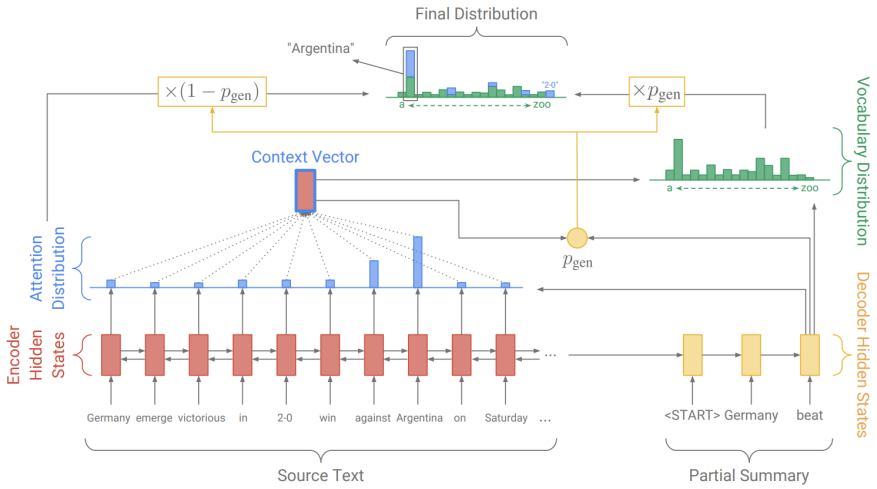

Figure 2: Pointer-generator model

described in the Motivation part.
The context vector is concatenated with the decoder state s_t and fed through two linear layers to produce the vocabulary distribution P_vocab:

$$P_vocab = softmax(V'(V[s_t, h_t^*] + b) + b'$$

During training, the loss for timestep t is the negative likelihood of the target word w_t^*:

$$loss_t = -logP(w_t^*)$$

The overall loss for the whole sequence is the average of them. One characteristics about the pointer-generator network above is that it allows both copying words via pointing, and generating words from a vocabulary. The generation probability $p_gen \in [0, 1]$ for timestep t is calculated from the context vector h_t^*, thus the decoder state s_t and the decoder input x_t is:

$$p_gen = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr})$$

In this way can we either choose to generate a word from the vocabulary or copying a word from the input sequence, following this distribution over the extended vocabulary:

$$P(w) = p_gen P_{vocab}(w) + (1 - p_{gen} \sum_{i:w_i=w} a_i^t)$$

3. **Challenge**
There are always some missing mappings between English and Shaltanacs, and

the way of tokenization and dependency relationship are not completely the same.
Also, there are relations between sentences. Thus, if we only do the split and combine on word level, we might not consider the effect between sentences or wrongly combine words that spanning different sentences.

2 Question Answering in Ewokese (20 points)

Congratulations! You have been hired by the Ewoks for building a QA system that would help them learn a bit about earthlings (say based on the English Wikipedia corpus). They want to be able to ask questions in Ewokese and have answers returned to them in Ewokese.

All they are willing to provide is about 100,000 sentences in Ewokese and a basic Ewok to English translation system which they believe is far from perfect (BLEU of 25.4 points) and a seed English to Ewok translation system (i.e. it can translate simple words and phrases without context).

(A) You wont be able to contact the Ewoks until you have built your system. So there is no possibility of obtaining any further training or annotation data. Provide a clear description of your QA system. Use a picture to describe the architecture. Tell us how you will train your system and justify your design choices. (15 points).

(B) If you could ask one Ewok to spend ten hours of their time helping you with annotations, what kind of annotation will you ask for, and why? Think about how many annotations you can reasonably expect in 10 hours. With this amount of data you have to argue that the specific annotations you are acquiring are the ones that will have the best impact on your system. (5 points)

Hint: Look up the idea of back translation. You can start here: `https://aclanthology.org/W18-2703/`.

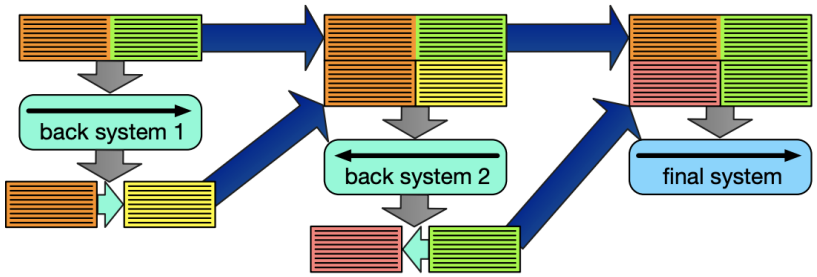

Figure 3: Re-Back-Translation

Algorithm 1 Iterative Back-Translation

Input: parallel data D^p, monolingual source, D^s, and target D^t text

Let $T_{\leftarrow} = D^p$
repeat
  Train target-to-source model $\Theta_{\leftarrow}$ on $T_{\leftarrow}$
  Use $\Theta_{\leftarrow}$ to create $S = \{(\hat{s}, t)\}$, for $t \in D^t$
  Let $T_{\rightarrow} = D^p \cup S$
  Train source-to-target model $\Theta_{\rightarrow}$ on $T_{\rightarrow}$
  Use $\Theta_{\rightarrow}$ to create $S' = \{(s, \hat{t})\}$, for $s \in D^s$
  Let $T_{\leftarrow} = D^p \cup S'$
until convergence condition reached

Output: newly-updated models $\Theta_{\leftarrow}$ and $\Theta_{\rightarrow}$

1. **Motivation**
   Since the Ewok corpus is very limited and we have a seed English to Ewok translation system, thus we consider use the idea of iterative back translation[Hoang et al., 2018] for text augmentation and training optimization. See Figure 3.
   We consider use a transformer-based LSTM representation for sentences.

2. **Training**
   Use the Iterative Back-Translation algorithm(see Figure 3) described in paper [Hoang et al., 2018], in which the target-to-source and source-to-target models we are to train represents question-to-answer and answer-to-question specifically.

3. **Manual Annotation**
   I would ask for annotations on some proper nouns which might very likely be important to the answers on the given sentences, So that we can reserve those nouns during back-translation.

3 Dependency Parsing (20 points)

Congratulations! Bloomberg just hired you because of your excellent background in NLP. They have been struggling with their dependency parser that they built for parsing news headlines.

Here are some examples:

1. `Stolen painting found by tree.`
2. `Eye drops off shelf.`
3. `Miners refuse to work after death.`

Propose a solution to develop a parser for headlines.

1. Collect five headlines that fail to parse correctly with a dependency parser that is trained on news stories but not on headlines. You can use `https://demo.allennlp.org/` as such a parser. Try to modify these sentences so that they convey the same meaning but are now proper sentences that parse correctly. Show the five headlines, the edited sentence you made and point out the reason based on this edit as to why the original headline did not parse. Summarize the reasons for these failures in terms of what we know about parsing. (5 points).

   Failed headlines:
   1) "Elon Musk offers timeline for Tesla Plaid deliveries to China" failed, because it could not recognize "Elon Musk". Modified as "The CEO of Tesla, Elon Musk, offers timeline for Tesla Plaid deliveries to China".
   2) "He fed her cat food" failed, it wrongly identifies "cat food" as a compound relation. It can be modified as as "He fed her cat the food".
   3) "The cat chased the mouse until it stumbled and fell", we could not infer from the sentence if the cat fell or the mouse fell. It can be modified as "The cat chased the mouse until it stumbled, and fell".
   4) "Each of us saw her duck", duck could be a noun or a verb, in more regular cases it identified as a noun. Modified sentence could be "Each of us saw her elude".
   5) "He saw a man on a hill with a telescope" failed due to the ambiguity of the prepositional phrase "saw" and "telescope". Its actual meaning should be the "man with telescope". Modified sentence as "He saw a man on a hill who is with a telescope".

2. Bloomberg's in-house journalists tell you that in most cases a headline has a corresponding sentence in the story – say the first sentence is always the expansion of the title written out in the standard language structure.

   For example, the first title above has the following corresponding first sentence:
   `Another stolen painting was found near a tree yesterday.`

   No Training Setup How will you use this information to improve the parsing of headlines if you had access to the first sentence of a story in addition to the

headline. You cannot retrain the parser but can obtain multiple candidate parses from it. (10 points).

1) Compare the parsing between headline and its multiple corresponding sentences. Rectify the dependencies between words and tagging of words according to the results of its multiple candidates.
2) We can also analyze the logic within these sentences through its parsing results, by transforming the dependency structure to logical forms such as lambda calculus, such that we can solve many ambiguous questions such as inference, propositional logic, vagueness, conjunctions, relative clauses through logical analysis[Reddy et al., 2016].

3. **Retraining Setup** Propose a way to retrain the parser given the same observation as above — you cannot annotate headlines or do any other manual annotation. You can assume that you have access to a large collection of (unlabeled) news headlines and their stories. (5 points).

   1) Constructs an intermediate representation of the input text intending to find salient content. Typically, it works by computing TF metrics for each sentence in the given matrix.
   2) Scores the sentences based on the representation, assigning a value to each sentence denoting the probability with which it will get picked up in the summary.
   3) Produces a summary based on the top k most important sentences. Some studies have used Latent semantic analysis (LSA) to identify semantically important sentences.

4 Machine Translation (20 points)

4.1 Translating headlines (5 points)

Suppose Google hires you for translating news headlines. Look at the following headlines:

- Key GOP lawmakers flip on health care after meeting.
- Spicer says part of spending bill will go to border wall.
- Ridge foster dad found not guilty of sex abuse charges.
- DWI suspect four times the legal limit with kids in car.
- Confident politician says he wants to 'prove them wrong' and get a Mideast peace deal.

Translate them through Google translate into a language other than English. What kind of errors do you observe? Describe the translation errors. Give three hypotheses that explains some of these errors. Now test each of your hypotheses by creating additional

headlines that also fail in a similar fashion. Provide some suggestions on what kind of information need to be modeled to address these errors.

* Depending on the target language, some translations might not fail. In such case, 1) make use of the other headlines that fail, or 2) pick another language where these headline translations would fail. For example, all of above headlines fail for Korean. 3) If none of the above works, you can pick other headlines of your choice from any news media. 4) If you only know one language, then you can use some native speaker of a different language to help you. If you choose to do so, note this in your answer clearly.

Observed error:
1) 'flip on' was not translated accurately, it uses its most basic meaning similar to 'turn to'.
2)'bill' which refers to 'an amount of money owed' but was translated as 'draft' here.
3) 'sex abuse charge' was not translated completely.
4) 'suspect' refers to 'advocate' here but was translated as 'doubt the genuineness or truth', 'four times' refers to an amount relation but was translated as a 'multiple' relation here. Thus 'the limit with kids in cars' was not correctly translated for sure.
5) Here the 'confident' should be of sarcastic meaning, but was translated into a positive term.
6) abbreviation such as 'GOP' and 'DWI' were not translated.

Hypothesis:
1) Google translate does not detect sarcasm or other semantics. Constructed test case: "I love that dress. The design really highlights your double chin."
2) Google translate lacks of common sense reasoning, thus when it comes to duplicate meaning words, it sometimes can not infer the right one. Constructed test case: "She always hates here, we all saw her duck last night."
3) It does not recognize and make the right translation of some proper nouns, similar situation can be found easily.

Suggestions:
1) Enhance its common sense reasoning and semantic analysis.
2) Enlarge its sense-tagged corpora.
3) Enhance its proper nouns translation by enlarging the vocabulary including their POS-tags.

4.2 Knowledge-backed Machine Translation (15 points)

News articles are often written for people from a specific region. This means often assumptions can be made about what people in the region would know about. For example, consider the following headline.

`Bloomberg announces presidential run.`

This works fine for the local audience in New York or even within the US but if this

headline were to be translated to Telugu, an Indian language, then a bit more context might be useful. The translation should ideally include some background information about Bloomberg. For example a translation should be the Telugu equivalent of the following sentence

`Bloomberg, the former NYC mayor and owner of Bloomberg News, announces presidential run.`

Design a knowledge-backed machine translation system on top of an existing seq2seq translation model to address this challenge.

Hint 1: Assume that you have access to knowledge bases that contains relations about entities, which are basically facts (e.g., mayor(Bloomberg, NYC), owner(Bloomberg, Bloomberg News)). This provides you a way to include useful information in the translation.

Hint 2: You will have to however think about what information is relevant for the current sentence. To figure out what is relevant information you can use some kind of distant supervision assumption. The idea of distant supervision was originally used for relation extraction.

Is there a way to find sentences about entities that contain additional background information about them?

Describe your solution in enough detail so that I can understand your ideas and solution. Please clearly specify the components in your model and how you will train them.

1. We need to identify the useful relational information defined in our background information in the sentences, thus we need Multi-Head Attention encoding.
2. We use Distant Supervision Generation (DSG) framework to generate a text that exactly described the given relation pairs. See Figure 4 [Fu et al., 2020].
   (a) It firstly trains an estimator to calculate each word's supportivenss with regard to the input data.
   (b) SE component is for estimating a supportiveness vector(SE) s which indicates whether each target word w_i is describing the input triple relation. It use self-supervised mechanism to train the model that maximize the margin between the target words' scroes and the negative sampled words' score.
   (c) Then use the pre-trained SE component to estimate a supportiveness vector s in both Sequence-to-Sequence training and generation(S2SG).
   (d) The RBS component combines s with the probability distribution of candidate words to obtain a better generation result.
   (e) f_K, f_T are the feature extraction components. $/hatT$ is the negative sample of the target.

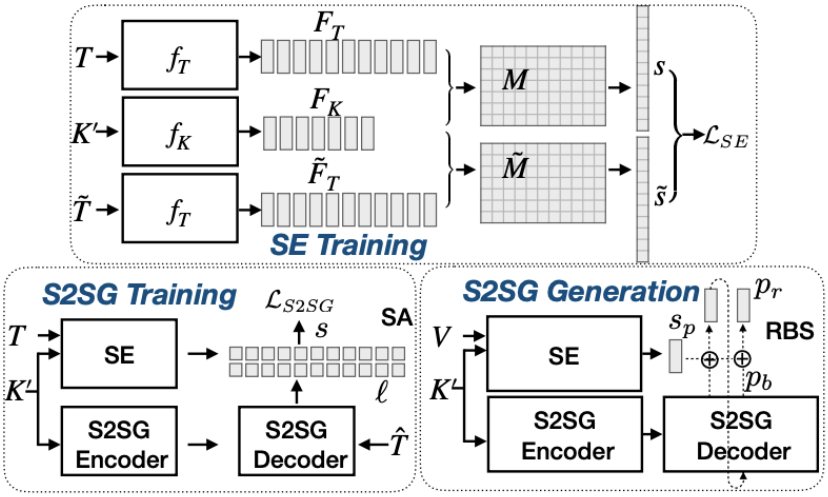

Figure 4: Distant Supervision Generation Framework

References

[Fu et al., 2020] Fu, Z., Shi, B., Lam, W., Bing, L., and Liu, Z. (2020). Partially-aligned data-to-text generation with distant supervision. *CoRR*, abs/2010.01268.

[Hoang et al., 2018] Hoang, V. C. D., Koehn, P., Haffari, G., and Cohn, T. (2018). Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

[Nguyen et al., 2017] Nguyen, D. Q., Dras, M., and Johnson, M. (2017). A novel neural network model for joint POS tagging and graph-based dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 134–142, Vancouver, Canada. Association for Computational Linguistics.

[Reddy et al., 2016] Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., and Lapata, M. (2016). Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

[See et al., 2017] See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.