CSC485 A1 Writeup

Richard Yan

October 2022

Family name: Yan

First name: Richard

Student #: 1005731193

Date: Oct. 7, 2023

I declare that this assignment, both my paper and electronic submissions, is my own work, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters and the Code of Student Conduct.

Signature: 

1 Transition-based dependency parsing

(a)

Step	Stack	Buffer	New dep.	Transition
0	[Root]	[To, ask, those, questions, is, to, answer, them]		
1	[Root, To]	[ask, those, questions, is, to, answer, them]		SHIFT
2	[Root, To, ask]	[those, questions, is, to, answer, them]		SHIFT
3	[Root, ask]	[those, questions, is, to, answer, them]	ask $\xrightarrow{mark}$ To	LEFT-ARC
4	[Root, ask, those]	[questions, is, to, answer, them]		SHIFT
5	[Root, ask, those, questions]	[is, to, answer, them]		SHIFT
6	[Root, ask, questions]	[is, to, answer, them]	questions $\xrightarrow{det}$ those	LEFT-ARC
7	[Root, ask]	[is, to, answer, them]	ask $\xrightarrow{dobj}$ questions	RIGHT-ARC
8	[Root, ask, is]	[to, answer, them]		SHIFT
9	[Root, is]	[to, answer, them]	is $\xrightarrow{csubj}$ ask	LEFT-ARC
10	[Root, is, to]	[answer, them]		SHIFT
11	[Root, is, to, answer]	[them]		SHIFT
12	[Root, is, answer]	[them]	answer $\xrightarrow{mark}$ to	LEFT-ARC
13	[Root, is, answer, them]	[]		SHIFT
14	[Root, is, answer]	[]	answer $\xrightarrow{dobj}$ them	RIGHT-ARC
15	[Root, is]	[]	is $\xrightarrow{xcomp}$ answer	RIGHT-ARC
16	[Root]	[]	Root $\xrightarrow{root}$ is	RIGHT-ARC

(b) A sentence with n words will be parsed in $2n$ parse step, because the parsing start with only the root in the stack, and all n words in the buffer. So for a completed parse, all the words will need to be shifted onto the stack (once per word), and to be connected to another word or the root via a dependency arc (also once per word), hence in total there will be $2n$ parse steps to complete the parse for a sentence of n words.

2 Graph-based dependency parsing

(a) Because when a crossing of edge happen in a non-projective tree, for example we have four words, name them root, a, b, c, where a is the beginning of the sentence, in which root $\rightarrow$ b, b $\rightarrow$ a, and a $\rightarrow$ c. In this case, the above algorithm can only handle the last two words of the stack (to create a dependency), so in order to handle word c, a and c must be the last two words on the stack, so b must already been processed (a dependent of another), but that only happened in the final step where root -¿ b, hence the algorithm will reach a dead end where no valid move is possible.

(b) train: 12081/12543 (96.3%)
dev: 1957/2001 (97.8%)
test: 2036/2077 (98.0%)

(c) Let's start with the first tree in q1, so I will write out a word on the left, followed by its descendants and gap degree. Gaps will be marked.

ROOT: 'ROOT To ask those questions is to answer them', gap degree: 0
To: 'To', gap degree: 0
ask: 'To ask those questions', gap degree: 0
those: 'those', gap degree: 0
questions: 'those questions', gap degree: 0
is: 'To ask those questions is to answer them', gap degree: 0
to: 'to', gap degree: 0
answer: 'to answer them', gap degree: 0
them: 'them', gap degree: 0
So the total gap degree of this tree is 0.

Now for the second tree in q2:
ROOT: 'ROOT John saw a dog yesterday which was a Yorkshire Terrier', gap degree: 0
John: 'John', gap degree: 0
saw: 'John saw a dog yesterday which was a Yorkshire Terrier', gap degree: 0
a dog: 'a dog | which was a Yorkshire Terrier', gap degree: 1
yesterday: 'yesterday', gap degree: 0
which: 'which', gap degree: 0
was: 'which was a Yorkshire Terrier', gap degree: 0
a Yorkshire Terrier: 'a Yorkshire Terrier', gap degree: 0
So the total gap degree of this tree is 1.

(e) We want to find a, b such that the uniform distribution $U(a, b)$ with mean $\mu = 0$ and variance $\sigma^2 = \frac{2}{m}$, where m is the number of inputs to our weight matrix W_A.

We have the system of equation:

$$\begin{cases} \frac{1}{2}(a+b) = 0 \\ \frac{1}{12}(b-a)^2 = \frac{2}{m} \end{cases}$$

$$a = -b$$

$$b = \sqrt{\frac{6}{m}}$$

So we have $a = -\sqrt{\frac{6}{m}}$ and $b = \sqrt{\frac{6}{m}}$, which yield $U(a, b)$ with mean 0 and variance $\frac{2}{m}$.

(g) Because in the arc scorer, it doesn't make sense to consider the appropriateness of a word to be a dependant when evaluating the appropriateness

of an edge, since every word will be a dependant (except ROOT) of some word. However, in the label scorer, we are evaluating the appropriateness of a label being on an edge, hence it make sense to consider the appropriateness of a word to be a dependant in a specific dependency relationship.

(h) Because we are using two neural networks to fulfill our desire, so we want the weight matrix to evaluate on both (i.e. we need to consider actually two classification problem, one is the appropriateness of the dependant word to be a dependant, the other one is the appropriateness of the head word to be a head).

(j) The reason we need to use a MST algorithm is because we have to make sure the 'graph' produced by selecting edges must form a tree, or otherwise if we only use argmax, it's possible that we only get a non-tree graph (for example a word will have multiple heads, etc.). But when we choose dependency relations, we are just picking one element (one relation) from a set of elements (set of all relations), so argmax is enough.

(l) Here is 3 examples where two parser don't agree (while both being not correct according to the corpus). For simplicity, I will just write down the sentence, and describe the part that are incorrect from the corpus.

First example:
Bush nominated Jennifer M. Anderson for a 15-year term as associate judge of the Superior Court of the District of Columbia, replacing Steffen W. Graae.

Edge-based parser from q1 marked 'Jennifer' and 'M.' as dependants of 'Anderson' (i.e. first and middle name as dependants of last name), while graph-based from q2 and the corpus annotate with the opposite (i.e. middle and last name as dependants of first name). Also edge-based parser look at those dependency as 'compound', while corpus say 'flat'.
The above situation also occur at the end, but now with edge-based parser giving last-¿middle-¿first (in contrast of both last -¿ middle and first). However graph-based parser are all over the place. The name in the end of the sentence is 'Steffen W. Graae', but the graph-based parser give Jennifer -¿ W. (i.e. mixing two names) and Court -¿ Graae as a 'nmod', so it's probably bad at looking at compounded pronouns I will say.
Edge-based parser say nominated -¿ judge with 'obl', while it should be term -¿ judge 'nmod'. I think maybe edge-based parser think this 'judge' as a verb probably?
Somehow graph-based parser think inside the phrase 'District of Columbia', District -¿ of, while it should be Columbia -¿ of.

Overall I think edge-based win on this, by not disturbing the meaning that much, while graph-based didn't perform so well on compounded pronouns I will say, and that's why I prefer edge-based more on this.

Second example:
The sheikh in wheel-chair has been attacked with a F-16-launched bomb.

Edge-based parser think in the phrase 'in wheel-chair', wheel -¿ in, while it should be chair -¿ in, since they all actually agree with that wheel is a compound dependant of chair.

In the phrase 'F-16-launched', it should be F and the second '-' -¿ launched, and the first '-' and 16 -¿ F. Both parser didn't do well on this, but graph-based agree on one more than edge-based.

Overall in this sentence, I think graph-based parser get a win, since it agree more with the corpus, while both didn't disturb the true meaning at lot. We can see that both parser doesn't do well in classify those complex compounded words in general.

Third example:
Today's incident proves that Sharon has lost his patience and his hope in peace.

There is only one word that both parser and the corpus disagree, while they agree on every other words and dependency relations. The word is 'hope' close to the end of the sentence.
Edge-base think lost -¿ hope with 'conj', while graph-based think lost -¿ hope with 'obj'. Corpus say patience -¿ hope with 'conj'. Since they agree on all other parts, I will probably say that according to my knowledge of English language and my understanding of this sentence, graph-based parser give the best result (even better than what corpus said). Since I think if we don't include the phrase 'his patience' in this sentence, '...lot his hope in peace' will be a totally fine sentence, while also didn't loss any information additionally with removal of 'his patience'. Thus I think patience and hope should be the same type of word in this sentence, in which only graph-based parser gives.