Introduction to Natural Language Syntax and Parsing: Practical

Ted Briscoe

Michaelmas Term 2021

The aim of this practical session is to evaluate and compare two or three parsers of your choice.

Resources

Output format: We suggest that you choose parsers which produce output in the style of Grammatical Relations (GRs) or directed Dependencies. The particular scheme is not that important; e.g. the CoNLL, RASP, Stanford or Universal Dependencies schemes would all be appropriate, but you should consider the extent to which the scheme utilised captures the information that a parser should recover (e.g. graphs vs. trees, interpretation of the GR types, etc).

Input sentences: A set of input sentences suitable for the evaluation can be found at the end of this handout. It is your responsibility to process these sentences so that they are in a suitable form for input to the parsers. **Choose at least 10 sentences for the analysis at least half of which should be from example number 15) onwards**, plus you may add your own sentences which highlight particular strengths or weaknesses of the parsers you have chosen. Some of the example sentences in the module handouts may be useful here.

It is often useful to simplify and/or modify sentences and try these if you suspect that a parser is failing or finding the wrong analysis because of sentence boundary detection, length, complexity, or through spurious interactions between constituents. When modifying the input, bear in mind that if you do this manually using your own linguistic insight you will inflate the perceived performance of the parser(s), so for objective evaluation you should preprocess the input into sentences and tokens with a (possibly customised) version of the scripts available in the tools you download, or with your own program.

Possible parsers: Parsers which you might evaluate include: the C&C CCG parser*; one of the Stanford NLP Tools parsers; the RASP system; SpaCy; the Berkeley parser; the Charniak parser; the Bikel parser; or the MALT or

Zpar dependencies-only systems, or an AMR parser if you are interested in and followed the formal semantics handout. The output of parsers which produce Penn Treebank trees can be converted to dependencies using the Stanford or other conversion scripts. It makes sense to choose parswers with different design characteristics and to compare how they perform in the light of these choices.

All of the parsers above, and the Stanford conversion script, can be found with a simple web search. A website which lists some NLP tools is:
http://nlp.stanford.edu/links/statnlp.html.
There are many others available, but we advise against choosing very obscure ones with little or no use by researchers other than the creators as they may be hard ot install and perform very differently from their description.
*Note on C&C: the Windows version is not as reliable as the Linux or Mac versions. Also note that many parsers have settings that will alter their accuracy; e.g. the RASP system has options to allow more than one PoS tag to be passed to the parser (-m), and to make use of verb subcat features (-s) to add trees to the default output ('-p -ogt') and to see all possible parses ('-p -n0'). (See $RASP/scripts/README for more information) and/or try:
% echo "I saw a man in the park with a telescope by the monument." | scripts/rasp.sh -p'-n0 -s' | more
to see an example of worst case ambiguity with multiple PPs (c.f. Section 2.5 of Handout 2).

Evaluation metrics: Parser evaluation in most published papers is either based on comparing Penn Treebank style constituency trees to the PTB gold standard using PARSEVAL, now deprecated, or more recently comparing dependency trees (occasionally graphs) to gold standards automatically derived from the Penn Treebank, DepBank, the universal dependencies treebanks, or the CoNNL dependency parsing shared task treebanks and associated evaluation scripts, e.g. MALTEVAL.

Here is a paper which describes how we think parser evaluation should be done:
http://aclweb.org/anthology/P06-2006
(see esp. Table 1, and think about macro vs micro F1 as a 'one number' informative measure). Another interesting approach is to construct a partial gold standard of 'hard' or 'important' dependencies and compare parsers' performance on this subset:
http://aclweb.org/anthology/D09-1085

In the past, some students have collaborated to produce (partial) gold standards for the test sentences. This is fine, but please acknowledge this in your reports, and ensure that your report is otherwise based only on your own work.

What You Need To Do

- Download a few of the available parsers; install them as necessary; and also read about their underlying probability / parse ranking models and search algorithms, output representations, etc. Choose two or three for comparison. (It is your responsibility to get all the parsers that you choose to compile and run on the machines you are using.)
- Investigate what pre-processing — in particular sentence boundary detection, tokenisation and POS tagging — is required for each parser.
- Examine the output of the parsers on the 10 input sentences of your choice (plus any additional sentences you have used) and see if the parsers make any errors. Can you make any generalisations about the errors each parser makes? Given what you know or can find out about the probability models and search algorithms adopted by each parser, can you explain why each parser makes the mistakes it does?
- As well as the qualitative evaluation above you may also compare parser output using a quantitative metric and construct a (partial) gold standard for some of the sentences in the data and compare against this.

Your Report

Your report should contain a concise summary and comparison of the errors made by each parser, and any general conclusions that you have been able to draw regarding the performance (accuracy and speed) of each parser. You may also want to discuss the strengths and weaknesses of your approach to evaluation as compared to the standard metrics used in the field.

Your report should not be longer than 5000 words and should include a word count. Your report should provide a pointer to a world-readable directory in your local filespace or on github that provides the complete output of the parsers that you ran and all your data as preprocessed by you.

Assessment

Your report will be graded out of 100 and will contribute 80% of the mark you receive for the module. Marks will be assigned for correctly identifying the errors made by each parser, for insightful comparison, for discussion of the issues of preprocessing, for generalisations concerning the parsing models and the types of errors observed, and for justification of the approach(es) taken to evaluation.

Data

(1) The old car broke down in the car park.

(2) At least two men broke in and stole my TV.

(3) The horses were broken in and ridden in two weeks.

(4) Kim and Sandy both broke up with their partners.

(5) The horse which Kim sometimes rides is more bad tempered than mine.

(6) The horse as well as the rabbits which we wanted to eat have escaped.

(7) It was my aunt's car which we sold at auction last year in February.

(8) The only rabbit that I ever liked was eaten by my parents one summer.

(9) The veterans who I thought that we would meet at the reunion were dead.

(10) Natural disasters – storms, flooding, hurricanes – occur infrequently but cause devastation that strains resources to breaking point.

(11) Letters delivered on time by old-fashioned means are increasingly rare, so it is as well that that is not the only option available.

(12) It won't rain but there might be snow on high ground if the temperature stays about the same over the next 24 hours.

(13) The long and lonely road to redemption begins with self-reflection: the need to delve inwards to deconstruct layers of psychological obfuscation.

(14) My wildest dream is to build a POS tagger which processes 10K words per second and uses only 1MB of RAM, but it may prove too hard.

(15) English also has many words of more or less unique function, including interjections (oh, ah), negatives (no, not), politeness markers (please, thank you), and the existential 'there' (there are horses but not unicorns) among others.

(16) Making these decisions requires sophisticated knowledge of syntax; tagging manuals (Santorini, 1990) give various heuristics that can help human coders make these decisions and that can also provide useful features for automatic taggers.

(17) The Penn Treebank tagset was culled from the original 87-tag tagset for the Brown Corpus. For example the original Brown and C5 tagsets include a separate tag for each of the different forms of the verbs *do* (e.g. C5 tag VDD for *did* and VDG tag for *doing*), *be* and *have*.

(18) The slightly simplified version of the Viterbi algorithm that we present takes as input a single HMM and a sequence of observed words $O = (o_1, o_2, ...o_T)$ and returns the most probable state/tag sequence $Q = (q_1, q_2, q_T)$ together with its probability.

(19) Thus the EM-trained "pure HMM" tagger is probably best suited to cases where no training data is available, for example, when tagging languages for which no data was previously hand-tagged.

(20) Coming home from very lonely places, all of us go a little mad: whether from great personal success, or just an all-night drive, we are the sole survivors of a world no one else has ever seen.

(21) Skill without imagination is craftsmanship and gives us many useful objects such as wickerwork picnic baskets. Imagination without skill gives us modern art.

(22) An MoD spokesman said: "Surveys of Astute have now been completed and she will proceed to Faslane under her own power. She is being escorted by tugs and HMS Shoreham."

(23) But far fewer people fully understand how the Media Lab operates, fits into MIT, and encourages such a creative environment; about half of the anniversary celebration's program focused on simply defining what the Media Lab is.

(24) Instead of constantly worrying about funding, the faculty and students can focus on their project, with the exception of sponsors' weeks, when they have to convince companies to start or continue their support.

(25) The doctors are warning that the NHS cannot make the £20bn of savings by 2014 that ministers expect, while simultaneously undertaking a huge reorganisation that will see England's 152 primary care trusts (PCTs) abolished and consortiums of GPs assume responsibility for the commissioning of services for patients.