Towards a better evaluation scheme for Universal Dependencies

Jimmy Callin
Uppsala University
jimmy.callin@gmail.com

Abstract

To be written.

1 Introduction

The role of language processing is becoming increasingly multi-lingual, which is reflected in recent efforts into providing dependency parsing frameworks that can reliably be applied on a multitude of languages. One of the most ambitious projects in this area is the Universal Dependencies (UD) framework (Nivre, 2015), where the goal is to create a parsing framework with a cross-linguistically consistent grammatical annotation. The purpose of this work is to remove the requirement of language specific components which has up to this point been a necessity due to inconsistent annotation standards.

To evaluate statistical parsing models, two of the most ubiquitous evaluation metrics are Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS). These have in their simplicity and intuitiveness served the research area well, but their design relies on a number of assumptions that we argue do not hold in the context of UD.

Firstly, the design of attachment scores assumes that we know next to nothing about the taxonomy and design choices of the parsing framework. This has historically been necessary since there have not been a consistently adapted framework for dependency parsing across many languages. We argue that recent progress in UD has made this assumption invalid. UD has a carefully specified framework that all UD treebanks has to adapt, and we can exploit this constraint to learn more about the performance of a model.

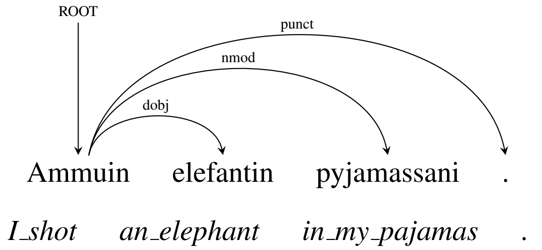

Figure 1: Finnish dependency tree for *I shot an elephant in my pajamas.*

Secondly, parsing results are becoming increasingly juxtaposed in a cross-linguistic manner, and this becomes especially true with the influence of UD. It is not uncommon to compare the output of e.g. English and Finnish under the assumption that equal evaluation scores is equivalent to equal parsing performance. The reason why this is problematic becomes apparent when studying grammatical morphemes in languages where these may be unbounded (i.e. function words) with languages where they typically are bounded on content words (i.e. affixes).

In figure 1, we have a Finnish sentence with four edges, while the equivalent sentence in English requires a total of eight edges[1]. This increase does not necessarily mean that the English sentence is harder to parse, since the additional edges are function words which are highly regular in their appear-

[1] Example taken from https://universaldependencies.github.io/docs/fi/overview/specific-syntax.html

Treebank	Token size
Arabic	282K
Basque	121K
Bulgarian	156K
Croatian	87K
Czech	1503K
Danish	100K
Dutch	200K
English	254K
Finnish	181K
Gothic	56K
Greek	59K
Hebrew	115K
Hindi	351K
Italian	252K
Norwegian	311K
Old Church Slavonic	57K
Persian	151K
Polish	83K
Portuguese	212K
Slovenian	140K
Spanish	423K
Swedish	96K

Table 1: Selected treebanks from the UD 1.2 treebank collection, with their token size.

ance in treebanks. The problem appears as soon as we introduce a parsing error into the trees: one faulty edge in the Finnish sentence would result in a performance reduction of 25%, while the same error in the English sentence only would reduce the accuracy by 12.5%.

Function words are regular in their appearance, and should therefore be relatively easy for a model to parse correctly. Our hypothesis is that languages with a comparatively large number of function words, like English, receive a more or less "free" performance boost when using classic evaluation metrics.

2 Related work

TODO

3 Data

Function relations	Content relations
aux auxpass case cc cop det expl mark neg mwe	acl advcl advmod amod appos ccomp compound conj csubj csubjpass dislocated dobj iobj list name nmod nsubj nsubjpass nummod parataxis remnant root vocative xcomp

Table 2: Classification of content and function relations.

We will be using a subset of the Universal Dependencies treebank 1.2. To keep the treebanks as consistent between each other as possible, we put a number of restrictions on what treebanks are allowed to join:

- They must all have morphological features.
- They must have at least 30K tokens.
- They must have a small ratio of non-projective trees.
- In the case of more than one treebank for a language, choose the one with manual corrections or largest token count

A total of 15 treebanks did not adhere to these restrictions. 5 of these had too few tokens, 4 lacked features, 2 had too many non-projective trees, while 4 treebanks were language duplicates. This leaves us with the languages listed in table 1. Most notably we lost the French and German treebanks.

4 Experimental setup

4.1 Determining function and content relations

We motivate our categorization of content and function relations based on the specification of universal dependency relations[2], linguistic intuition, and a newly developed statistical method. Firstly, let us define what we consider to be content and function relations:

- A *content relation* is a relation that links a content word with another content word.
- A *function relation* is a relation that links a word with a function word.

[2] http://universaldependencies.github.io/docs/u/dep/index.html

4.2 Categorization by specification

Given the previous definition as well as the specification of universal dependency relations, we can categorize a relation based on how it should occur in UD treebanks without knowing if the treebanks strictly adhere to the rules set up by the framework. Going through each dependency relation in this manner we ended up with a classification as presented in table 2.

We chose to remove some relations where we cannot make assumptions of its content, labeled *other*. The *foreign* relation has no restrictions of what type of word it should choose as a dependent as long as it is a foreign word. *List* are used in cases where the content cannot be easily analyzed. *Reparandum* and *dep* are neither of semantic nor syntactic nature and we cannot make any assumptions of their content. *Punct* and *discourse* are removed for similar reasons, but also due to particles often being ignored in other evaluation schemes.

4.3 Placing dependency relations on a function–content spectrum

Next question to answer is if we can motivate our classification not only on linguistic intuition and the specification of dependency relations, but also from an empirical perspective. We do this by adhering to our previous definition on function dependencies, and what we know of the nature of function words. Since they are part of closed word classes, meaning new words rarely are introduced into their categories, we can expect the number of distinct word types to be quite small, especially when compared to the word classes shared by typical content words such as *nsubj*.

Assuming this holds, we expect that the probability of a word given a function relation to be zero for all but a few cases, while the probability of a word given a content relation should be much more evenly spread. This can be quantified by measuring a relation's entropy given its word probabilities. We call this measure Word Entropy (WE). Calculated for all treebanks we get the averaged WE.

Here we formally define word entropy. A probability distribution p takes as input a word w conditioned on a treebank $t \in T$, and a dependency relation r. H is the entropy function that takes $p(w|r,t)$ as input and calculates its entropy. The entropy function is normalized by its upper bound $\log n_w$, where n_w is the size of the vocabulary. This keeps the range of the function to $[0, 1]$. To calculate the WE for a set of treebanks, we average WE for all treebanks $t \in T$. In the case a dependency relation is not present in a treebank, we set its WE to 0.5 to imply that it is neither a content nor a function relation.

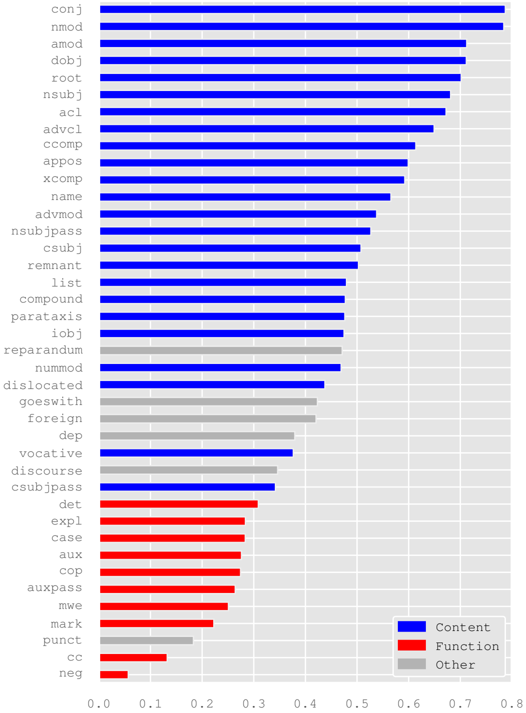

Figure 2: Averaged word entropy for all universal dependency relations, with the manually created categories.

This gives us the following mathematical functions:

$$\mathrm{WE}(r,t) = \frac{H(p(w|r,t))}{\log n_w}$$

$$\text{Averaged WE}(r,T) = \frac{1}{n_T} \sum_{t \in T} \mathrm{WE}(r,t)$$

The averaged WE for UD 1.2 is presented in figure 4.3, along with our manual categorization of dependency relations. Ignoring the *other* relations, we find that the word entropy gives an intuitive ordering of the dependency relations, while empirically supporting our choice of content and function relations.

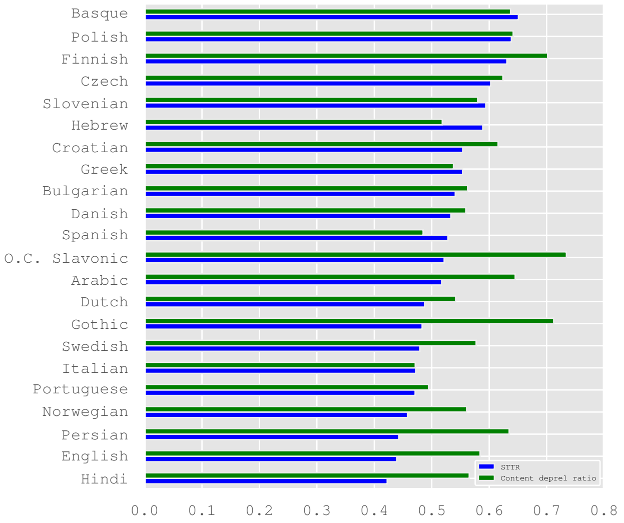

Figure 3: Standardized type/token ratio for chunks of 1000 tokens (blue) and content dependency (green). $R = 0.27$

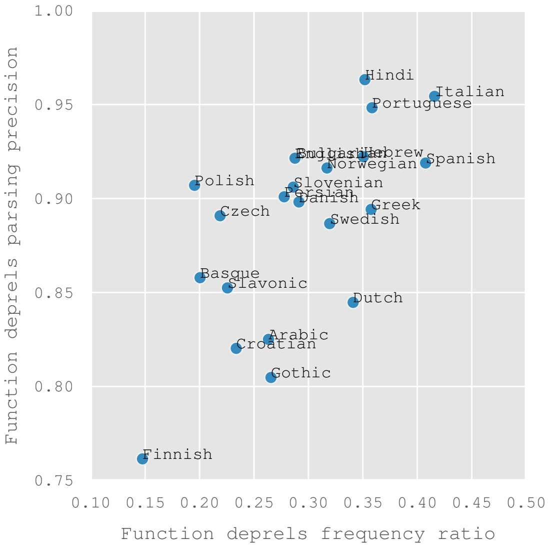

Figure 4: Precision of functional dependency relations, and the frequency ratio of functional dependency relations for each language. $R = 0.67$

4.4 Finding correlation with external degree of synthesis measurements

We would expect the ratio of function words in a given language's treebank to correlate with its degree of synthesis. Measuring degree of synthesis is not a trivial task, and there have been several proposed algorithms for this. Despite having obvious drawbacks, an often used indirect measurement of degree of synthesis is the type/token ratio. This assumes that synthetic languages, with their morphologically rich systems, will have fewer tokens per word than analytic languages such as English or Hindi. This is not particularly robust when comparing across corpora of different sizes. As such, we will be using the *standardised* type/token ratio (STTR), which calculates average TTR in chunks of 1000 tokens[3]. Figure 4.3 shows that there is a weak correlation between languages' frequency ratio of function relations and their STTR. Some languages are clear outliers such as Old Church Slavonic, Arabic, Gothic, Persian, English, and Hindi. Whether this is a result of a bad degree of synthesis measurement, a bad classification of function dependency relations, or a combination of both is up for discussion.

4.5 Result of classification

In 2, we list all dependency relations according to their classification as previously motivated.

By studying figure 4, we see that there is a clear correlation between the frequency ratio of function relations in a language with its precision of the same relation class. What is interesting is that this does not hold when looking at content relations, as seen in figure 5. This suggests that languages with a high degree of function dependency relations has an unfair advantage when comparing attachment scores across languages.

5 Analysis of results

Figure **??** shows how the variance for high performing languages has decreased, while the overall variance has not been affected when looking at the whole language spectrum.

One hypothesis for why we do not see a relative decrease that is correlated to their sTTR values is that the function words works better as a structure to make a better classification for the content words.

[3] Introduced by Mike Scott in `http://lexically.net/downloads/version6/HTML/index.html?type_token_ratio_proc.htm`

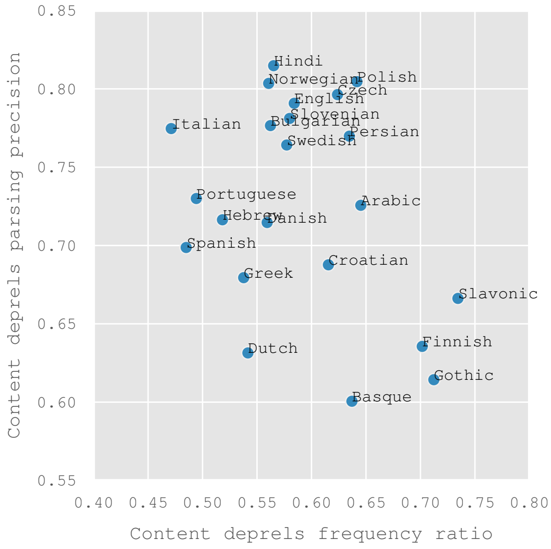

Figure 5: Precision of content dependency relations, and the frequency ratio of content dependency relations for each language. $R = -0.33$

6 Acknowledgements

We thank Jörg Tiedemann for allowing us to use his parsing output on UD 1.0 for initial experiment development.

7 References

Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing. In *Computational Linguistics and Intelligent Text Processing*, pages 3–16. Springer, editions.

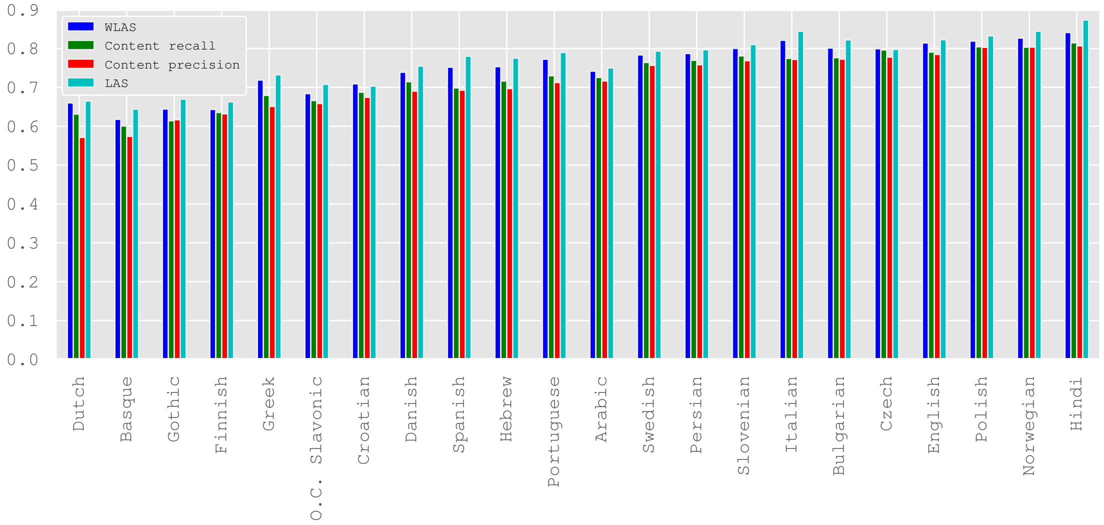

Figure 6: Overall LAS score, precision and recall for content dependencies.

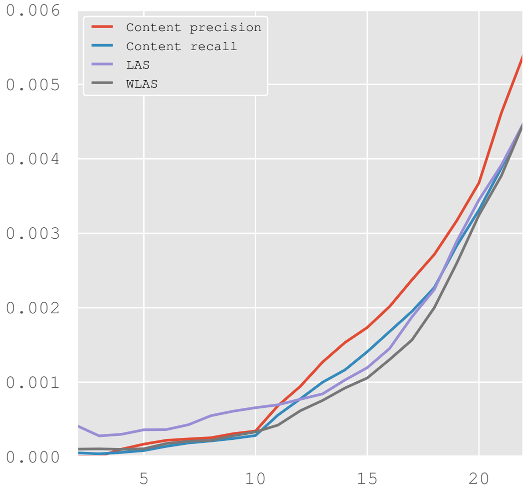

Figure 7: Cumulative variance when adding languages in a top-scoring order.