FastKASSIM: A Fast Tree Kernel-Based Syntactic Similarity Metric

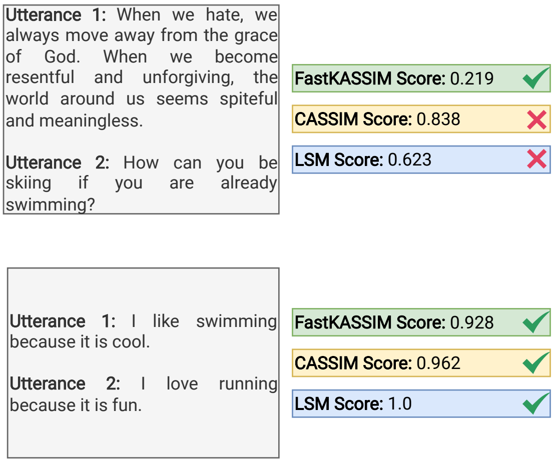
Maximillian Chen,[*] Caitlyn Chen,[*] Xiao Yu,[*] Zhou Yu
Department of Computer Science, Columbia University, New York, NY
maxchen@cs.columbia.edu
{caitlyn.chen, xy2437, zy2461}@columbia.edu

Abstract

Syntax is a fundamental component of language, yet few metrics have been employed to capture syntactic similarity or coherence at the utterance- and document-level. The existing standard document-level syntactic similarity metric is computationally expensive and performs inconsistently when faced with syntactically dissimilar documents. To address these challenges, we present FastKASSIM, a metric for utterance- and document-level syntactic similarity which pairs and averages the most similar constituency parse trees between a pair of documents based on tree kernels. FastKASSIM is more robust to syntactic dissimilarities and runs up to to 5.32 times faster than its predecessor over documents in the r/ChangeMyView corpus. FastKASSIM's improvements allow us to examine hypotheses in two settings with large documents. We find that syntactically similar arguments on r/ChangeMyView tend to be more persuasive, and that syntax is predictive of authorship attribution in the Australian High Court Judgment corpus.

1 Introduction

Syntax, the form of language, plays a crucial role in all aspects of natural language and communication, whether explicitly or implicitly. In storytelling, writers often have their own styles rooted in different syntactic tendencies (Feng et al., 2012), allowing syntax to become indicators in prediction tasks such as gender (Sarawgi et al., 2011) and authorship (Raghavan et al., 2010) attribution. Syntax also has social connotations in different cultures — for example, in Russia, different social and demographic groups tend to use different syntactic patterns (Bogdanova-Beglarian et al., 2016). Such examples makes syntactic consistency crucial to capture in tasks such as machine translation and dialogue generation so that social conventions are not lost. Yet, recent research focuses primarily on evaluating similarity and coherence in terms of dimensions like semantics, the meaning behind language, (with approaches such as BERT embeddings (Reimers and Gurevych, 2019; Zhang et al., 2019b)), or lexical overlap (e.g., BLEU (Papineni et al., 2002)) — even in work which uses syntax as an input to improve translation quality (Zhang et al., 2019a). A lack of work on syntax can be partially attributed to the absence of a practical, efficient metric that specifically compares syntax at the utterance level. The current standard metric is CASSIM (Boghrati et al., 2016, 2018), but CASSIM uses a computationally expensive distance metric and can yield inconsistencies when comparing syntactically dissimilar documents (e.g., Figure 1). To address this issue, we introduce *FastKASSIM*, a *Fast Tree Kernel-bAsed Syntactic SIMilarity Metric*[1], an improved metric for syntactic similarity at the utterance- and document-level.



Utterance 1: When we hate, we always move away from the grace of God. When we become resentful and unforgiving, the world around us seems spiteful and meaningless.

Utterance 2: How can you be skiing if you are already swimming?

FastKASSIM Score: 0.219 ✓
CASSIM Score: 0.838 ✗
LSM Score: 0.623 ✗

Utterance 1: I like swimming because it is cool.

Utterance 2: I love running because it is fun.

FastKASSIM Score: 0.928 ✓
CASSIM Score: 0.962 ✓
LSM Score: 1.0 ✓

Figure 1: Comparison of FastKASSIM, CASSIM, and Linguistic Style Matching similarity scores. Top: two dissimilar utterances. Bottom: two similar utterances. All three metrics have strong agreement in cases of similar syntactic structure, but only FastKASSIM is able to recognize syntactically dissimilar utterances. The parse trees of these examples are visualized in Appendix D.

[*] denotes equal contribution.

[1] https://github.com/jasonyux/FastKASSIM

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 211–231
May 2-6, 2023 ©2023 Association for Computational Linguistics

Like its predecessor, CASSIM, FastKASSIM computes the constituency parse tree of each sentence in a pair of documents, and the similarity between each pair of parse trees. But, while CASSIM used Edit Distance (Pawlik and Augsten, 2011; Zhang and Shasha, 1989) for similarity, we propose using a *Label-based Tree Kernel* (henceforth LTK), our more syntactically thorough implementation of the Fast Tree Kernel (Moschitti, 2006). We evaluate FastKASSIM against CASSIM and Linguistic Style Matching (henceforth LSM; Niederhoffer and Pennebaker (2002); Ireland and Pennebaker (2010)). We find that FastKASSIM is more robust in cases of dissimilarities between documents and is generally more agreeable with human perception of differences in syntax. Additionally, the runtime of LTK is much faster than that of Edit Distance; it *scales linearly with the number of node pairs with the same label* in a pair of parse trees. We empirically show large improvements in runtime with FastKASSIM.

Previously, it was difficult to observe the role of syntax in behavioral phenomena at scale due to runtime constraints. Here, we contribute a study of hypotheses in two sets of applications. First, we examine the relationship between the persuasiveness of online arguments and syntactic similarity, and second, we observe the viability of syntax as an indicator in authorship attribution. FastKASSIM unlocks potential for evaluatory use in more contexts where it is important to preserve syntactic consistency and writing style, e.g., style transfer, machine translation, and story generation.

2 Related Work

A few early studies focused solely on capturing syntactic structures. Sagae and Gordon (2009) sought to cluster *words* by syntactic similarity. In order to establish a distance metric, they computed the cosine distance between vector representations of their unique constituency parses. Other approaches have used LIWC (Tausczik and Pennebaker, 2010). Danescu-Niculescu-Mizil et al. (2011) took a probabilistic approach to measure symmetry and influence of linguistic style.

Other early analytical work found that people will adjust their syntax to match dialog systems' syntactic (Stoyanchev and Stent, 2009) and lexical (Stoyanchev and Stent, 2009; Hoshida et al., 2017) choices. Reitter et al. (2006) found that individual syntactic productions would repeat at low "distances" across utterances in both task-oriented and “spontaneous” dialog. For instance, Reitter and Moore (2007) found that syntactic priming was predictive of success on the HCRC Map Task (Anderson et al., 1991). Baker et al. (2021) similarly discussed the use of syntactic similarity and overall linguistic style synchrony as an indicator of trust and cohesion in teamwork settings, and Boncz (2019) used syntactic similarity in modeling cognitive alignment.

Syntactic features have been shown to improve prediction performance in downstream tasks, e.g., authorship attribution (Posadas-Duran et al., 2014; Raghavan et al., 2010; Zhang et al., 2018) and gender attribution (Sarawgi et al., 2011). In each case, the studies found significant performance gains from models that included syntax features. Despite interest in syntax and clear improvements in prediction performance, the vast majority of recent work primarily focuses on semantic, or even lexical similarities/differences. This ranges from traditional methods such as TFIDF or Jaccard similarity to modern approaches including BERT embeddings (Devlin et al., 2019; Reimers and Gurevych, 2019; Zhang et al., 2019b) and AMR kernels (Opitz et al., 2021). Some approaches use syntactic features specific to certain domains such as Twitter (Alnajran, 2019; Little et al., 2020) or web documents (Broder et al., 1997; Pereira and Ziviani, 2003). Other metrics include “syntactic elements” which take on various forms of parts-of-speech aggregation (Alnajran, 2019; Pakray et al., 2011).

A *syntactic* similarity metric should appropriately consider differences in syntactic structure at the word-, utterance-, and document-level, as opposed to aggregating parts-of-speech or relying on domain-specific features. To our knowledge, CASSIM is the only metric to do this and has been proposed as a solution in applications ranging from measuring communicative alignment (Boncz, 2019; Baker et al., 2021) to evaluating stylistic creativity in language learning (Kokkola and Rydström, 2022) to clustering text (Boghrati et al., 2017). However, CASSIM relies on the expensive Edit Distance metric, and occasionally assigns high similarity scores to documents that appear syntactically dissimilar. An improved syntactic similarity metric would afford new opportunities, from creating novel syntax feature vectors for classification tasks (e.g. authorship and gender attribution), to measuring syntactic coherence in machine translation.

Algorithm 1 FastKASSIM

```
DOCUMENTS $D_1, D_2$
for sentences $S_1, S_2$ in $D_1, D_2$ do
    Compute Parse Tree($S_1$), Parse Tree($S_2$)
end for
for Parse Trees $P_1, P_2$ do
    Compute Tree Kernel:
    $s \leftarrow 0$
    for Node Pair $n_1, n_2$ in $P_1, P_2$ do
        $s \leftarrow s + \Delta_{lb}(n_1, n_2)$
    end for
    Kernel $\leftarrow$ normalize($s$)
end for
Hungarian Algorithm Max. Cost Assignment
return mean(maximal cost pairings)
```

3 FastKASSIM

3.1 CASSIM Background

CASSIM (Boghrati et al., 2018) was the first metric to compute syntactic similarity at the document-level. Their original algorithm uses the Stanford Parser (Klein and Manning, 2003; Chen and Manning, 2014) to compute the parse tree for each sentence in a pair of documents, before computing the Edit Distance (Wagner and Fischer, 1974) between each parse tree pairing. Then, they construct a bipartite graph and use the Hungarian Algorithm for minimum cost assignment (Kuhn, 1955) to pair each tree in one document to the lowest distance tree in the second document. They finally average the Edit Distances of the minimal cost pairings. When there are different numbers of sentences, the number of assignments will correspond to the number of sentences in the document with fewer sentences. Each of that document's sentences will get paired with the most similar sentence in the second document, and the least similar sentences in the second document will remain unpaired. The final Edit Distance between a pair of parse trees P_1, P_2 is normalized as $\frac{EditDistance}{Size(P_1)+Size(P_2)-2}$, where $Size(P)$ is the number of nodes in P.

An important advantage of CASSIM is that it is generalizable to any corpus; it does not represent syntax using platform-specific features like Alnajran (2019); Little et al. (2020). However, the cost of exhaustively using a metric such as Edit Distance is rather penalizing, as its implementations range in asymptotic time complexity from $\Theta(mn)$ (Wagner and Fischer, 1974) to $O(s \times min(m, n))$ (Ukkonen, 1985), where m and n are the string lengths, and s is the maximal Edit Distance.

Algorithm 2 Delta$_{lb}$ Function (Δ_{lb})

```
Tree Nodes $n_1, n_2$; $cache$
Decay $\lambda$; Subtree/Subset Tree Indicator $\sigma$
if $n_1, n_2$ is cached then
    return cache($n_1, n_2$)
end if
if $n_1, n_2$ have different labels then
    return 0
end if
if both $n_1, n_2$ are preterminals then
    cache($n_1, n_2$) $\leftarrow \lambda$ if same label, else 0
    return cache($n_1, n_2$)
end if
Product $\leftarrow$ 1
for child $c_1$ of node $n_1$ do
    Accumulator $\leftarrow$ 0
    for child $c_2$ of node $n_2$ do
        Acc. $\leftarrow$ Acc. + Delta$_{lb}$($c_1, c_2$)
    end for
    Product $\leftarrow$ Product $\times(\sigma+$ Acc.)
end for
cache($n_1, n_2$) $\leftarrow \lambda\times$ Product
return $\lambda\times$ Product
```

3.2 The FastKASSIM Algorithm

In large multi-sentence documents, repeated Edit Distance becomes the most expensive component of CASSIM. Thus, we propose FastKASSIM, which avoids the expensive Edit Distance computation by using Tree Kernels (Moschitti, 2006). Tree Kernels can greatly reduce time complexity by caching between recursive subcalls. The Fast Tree Kernel algorithm Moschitti (2006) runs in *linear time on average with respect to parse tree sizes.*

We propose FastKASSIM, which replaces CASSIM's Edit Distance with a new normalized Tree Kernel. Figure 2 provides a high-level overview of FastKASSIM, which is formally described in Algorithm 1. However, the Fast Tree Kernel does not allow for the case in which two parse tree nodes have matching labels but different productions. We thus also introduce the Label-based Tree Kernel (LTK)[2], which compares the labels at each node in a pair of subtrees or subset trees[3]. This also

[2] There is a very strong correlation between LTK and the Fast Tree Kernel ($R = 0.97$, p<0.001).

[3] Moschitti (2006) defines a subtree as a node and all its

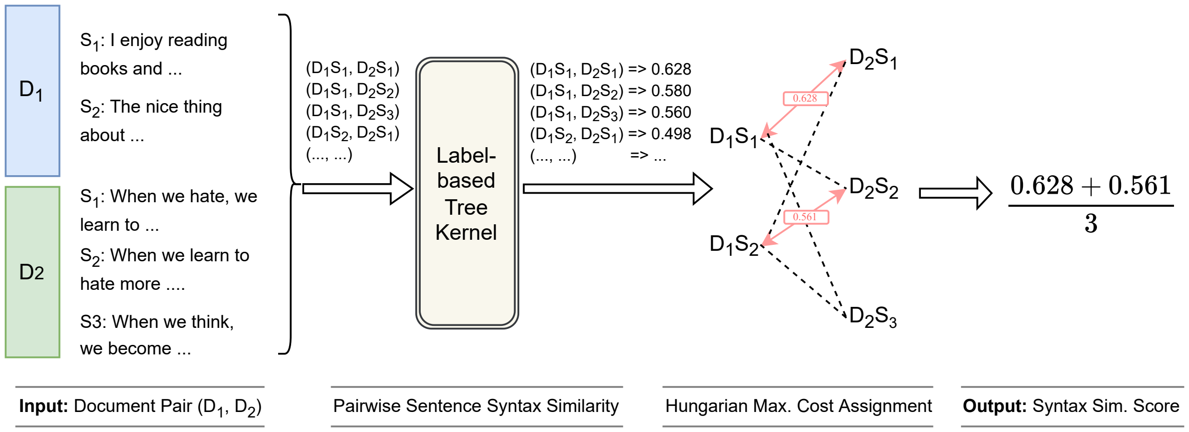

D$_1$
S$_1$: I enjoy reading books and ...
S$_2$: The nice thing about ...
D2
S$_1$: When we hate, we learn to ...
S$_2$: When we learn to hate more ....
S3: When we think, we become ...
(D$_1$S$_1$, D$_2$S$_1$)
(D$_1$S$_1$, D$_2$S$_2$)
(D$_1$S$_1$, D$_2$S$_3$)
(D$_1$S$_2$, D$_2$S$_1$)
(..., ...)
Label-based Tree Kernel
(D$_1$S$_1$, D$_2$S$_1$) => 0.628
(D$_1$S$_1$, D$_2$S$_2$) => 0.580
(D$_1$S$_1$, D$_2$S$_3$) => 0.560
(D$_1$S$_2$, D$_2$S$_1$) => 0.498
(..., ...) => ...
D$_1$S$_1$
D$_1$S$_2$
D$_2$S$_1$
D$_2$S$_2$
D$_2$S$_3$
0.628
0.561
$\frac{0.628 + 0.561}{3}$
Input: Document Pair (D$_1$, D$_2$)
Pairwise Sentence Syntax Similarity
Hungarian Max. Cost Assignment
Output: Syntax Sim. Score

Figure 2: A high-level illustration of FastKASSIM computation. The parse trees of all sentence pairs between D_1, D_2 are computed using LTK. The Hungarian algorithm is used to pair together the most similar parse trees of each sentence in the two documents by the “maximal cost” (i.e., the largest tree kernels). The score is normalized by summing the paired kernel values then dividing by the number of sentences in the document with more sentences. D_2S_3 is unpaired because D_1S_1, D_2S_1 and D_1S_2, D_2S_2 are paired, and D_2 has more sentences than D_1.

more closely follows (Collins and Duffy, 2002), which proposes comparing the actual subset trees rooted at two nodes in each parse tree, rather than the production. Figure 3 depicts the LTK algorithm computing the number of shared subset trees in a pair of parse trees. More formally, as described in lines 7-11 of Algorithm 1, LTK accumulates the value of Δ_{lb} (Algorithm 2), which is the number of common fragments rooted in a pair of parse tree nodes n_1, n_2.

We follow Moschitti (2006) by normalizing $LTK(T1, T2)$ as $\frac{LTK(T1,T2)}{\sqrt{LTK(T1,T1)\times LTK(T2,T2)}}$. This normalized tree kernel is not biased towards tree shape, in contrast to CASSIM’s normalized Edit Distance ($\frac{EditDistance}{Size(P_1)+Size(P_2)-2}$). Under CASSIM’s normalization, if two sentences resulted in the *same parse tree size despite being composed of entirely different labels*, the normalized Edit Distance approaches 0.50 (the expression approximates $\frac{Size(P_1)}{2\times Size(P_1)-2}$). In other words, according to CASSIM, sentences with the same shape but different parts-of-speech should be neither similar nor dissimilar. We further highlight possible examples of this bias by visualizing parse trees in Appendix D. Ultimately, our normalized LTK results in significant runtime improvements over Edit Distance, as we show in Section 3.3 and derive in Appendix A, and agrees strongly with human perception, as we show in Section 4.

Like the original CASSIM algorithm, our high-level algorithm allows for flexibility in the choice of which parser to use, allowing for future improvements in runtime and correctness as research in parsing progresses. FastKASSIM similarly allows for flexibility in the implementation of tree kernels. Our implementation will be publicly released upon acceptance. In order to directly compare FastKASSIM and CASSIM, we default to using the Stanford Parser (Chen and Manning, 2014) and LTK, our aforementioned modified approach to the Fast Tree Kernel (Moschitti, 2006).[4]

3.3 Overall Metric Runtime Comparison

The largest difference in the runtime of CASSIM and FastKASSIM is that CASSIM uses a normalized Edit Distance to evaluate parse tree similarity, while FastKASSIM uses a normalized tree kernel.

LTK recursively computes Δ_{lb} across all n_1, n_2 pairs in parse trees P_1, P_2. But, importantly, *all comparisons are cached to avoid repetition.* This results in LTK having an asymptotic runtime complexity of $O(S_{12})$, where S_{12} is the total number of pairs of nodes in a pair of parse trees P_1, P_2 that have the same label. We prove this runtime in Appendix A. This is a large improvement over Edit Distance’s runtime complexity of $O(s \times min(m, n))$. We confirmed that these asymptotic improvements apply to real-world uses cases by comparing how Edit Distance and LTK scale with the product of parse tree sizes in Figure 6 of the Appendix, finding that LTK scales sublinearly while Edit Distance scales superlinearly.

descendants, whereas a subset tree does not require its leaves to be terminal.

[4] However, we provide users with a native interface to interchangeably use any parser supported by NLTK (Bird et al., 2009).

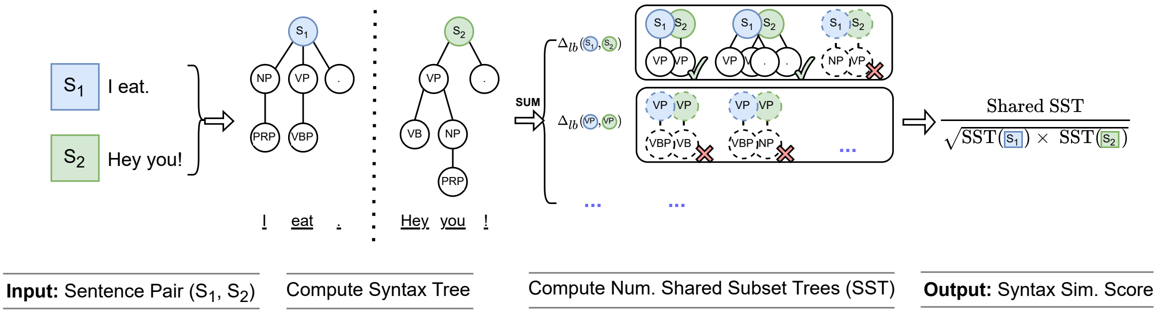


Figure 3: Overview of the Label-based Tree Kernel. The parse trees of a pair of sentences are computed, along with the number of common fragments rooted in each pair of parse tree nodes (i.e., number of shared subset trees). This is normalized by dividing by the square root of the product of the number of subset trees in each parse tree.

While Figure 6 indicates that LTK can be up to an order of magnitude faster than Edit Distance, the largest bottleneck in overall time is still the time to compute each parse tree. Thus, in Figure 4, we investigated the difference in "end-to-end" runtime between FastKASSIM and CASSIM without precomputing the parse trees.

In this experiment, the ChangeMyView dataset (henceforth CMV; Tan et al. (2016)) is used, providing a corpus of unstructured text with large document sizes, to evaluate the promises of FastKASSIM and CASSIM for their abilities to process entire documents. First, we sample entire document pairs and record the time it takes to compute the syntactic similarity of each pair. Each pair is randomly sampled from the 18,363 posts in the CMV training set. Then, we exhaustively paired documents based on the product of their document sizes, providing an approximation of the number of comparisons between parse trees. The document length for each CMV root posts has high variance, so document length products are grouped into bins. For each bin, we randomly sample 60 document pairs and report the average runtime.

Figure 4 shows that FastKASSIM scales well in runtime as the product of document lengths increases. For instance, when syntactic similarity between documents of lengths 300 words and 310 words were compared (product of $93,000$), CASSIM needed on average 113.3 seconds while FastKASSIM took only 21.3 seconds on average. Given these drastic improvements in time complexity, it is now more feasible to compute syntactic similarity at the document level for large corpora.

4 Evaluating FastKASSIM

In this section, we first demonstrate FastKASSIM's overall ability to differentiate between similar and dissimilar documents. Then, we correlate its scores with CASSIM and LSM. Finally, we discuss FastKASSIM's advantages by explaining discrepancies in scoring.

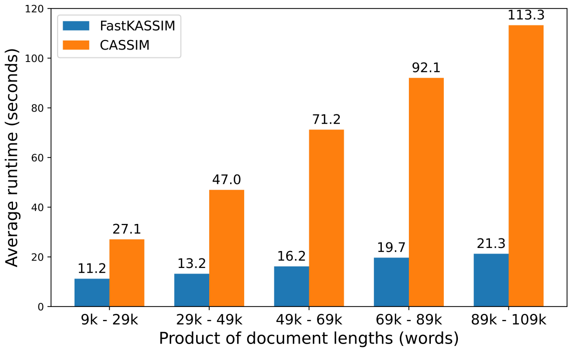


Figure 4: Runtime comparison between FastKASSIM and CASSIM. On the CMV corpus, FastKASSIM runs 2.42 times to 5.32 times faster on average, depending on document size.

4.1 Discriminating Between Syntactically Similar and Dissimilar Documents

Boghrati et al. (2018) validated CASSIM by comparing whether it was consistent with human perception of syntactic similarity. The authors asked Mechanical Turkers to write syntactically similar sentences given a sentence prompt. This resulted in a dataset of 472 English documents from 118 anonymous human annotators, and the authors found that CASSIM was able to "identify syntactically similar documents."

Following Boghrati et al. (2018), we computed the syntactic similarity between each pairing of sentences generated both within the same prompt and between different prompts. Each individual prompt is structurally quite different (Table 1). By construction, *documents resulting from the same prompt should be syntactically similar, whereas documents resulting from different prompts should*

Prompts
1. The two most important days in your life are the day you are born and the day you find out why. The nice thing about being a celebrity is that you bore people and they think it's their fault.
2. I am enough of an artist to draw freely upon my imagination. Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.
3. When we love, we always strive to become better than we are. When we strive to become better than we are, everything around us becomes better too.
4. What is the point of being alive if you don't at least try to do something remarkable?

Table 1: Prompts for the crowd-sourced corpus collected by Boghrati et al. (2018).

be dissimilar. As per their work, we fit a maximal structure linear mixed effect model with an indicator for whether the sentence corresponded to the same prompt or a different prompt as a fixed effect and the document ID as a random effect against standardized syntactic similarity[5].

We computed ANOVA of this full model against the reduced model, which drops the comparison type indicator as a fixed effect. The ANOVA χ^2 test on the impact of comparison type yields statistically significant differences in the distribution of FastKASSIM ($\chi^2 = 1438.9$, $p < 0.001$), CASSIM ($\chi^2 = 331.84$, $p < 0.001$), and LSM ($\chi^2 = 201.85$, $p < 0.001$) scores between syntactically similar and dissimilar documents. FastKASSIM results in the largest effect size, 1438.9, indicating it creates the largest differences in distribution.

4.2 Correlating Syntax Metrics

LSM[6] is a metric computing similarities from function word categories. This has ties to matching syntax, as those matching function words correspond to specific parts-of-speech. Moreover, LSM is a widely accepted metric for synchrony and correspondence of general linguistic style in documents (e.g. Chartrand et al. (2005); Ludwig et al. (2013)). We examine the actual similarity scores calculated in the previous section on the crowd-sourced document similarity corpus collected by Boghrati et al. (2018) using each of LSM, FastKASSIM, and CASSIM. In Figure 5, we see that there is a moderately strong correlation between FastKASSIM and LSM ($R = 0.5$, $p < 0.001$). This indicates that FastKASSIM is able to detect matches in key parts-of-speech. We would not expect to see a greatly higher correlation, because LSM is a measure of function words rather than a holistic measure of syntax. On the other hand, while we see a statistically significant correlation between CASSIM and LSM, its correlation coefficient is much lower ($R = 0.11$, $p < 0.001$), indicating a smaller connection between CASSIM representations and function words. This is likely due to the biased Edit Distance normalization mentioned in Section 3.1. Moreover, we actually find an overall negative correlation between FastKASSIM and CASSIM ($R = -0.33$, $p < 0.001$) with a seemingly bipartite relationship. There is an apparent disagreement over documents that FastKASSIM deems dissimilar, with agreement over documents that FastKASSIM deems similar.

4.3 Discrepancies Across Syntax Metrics

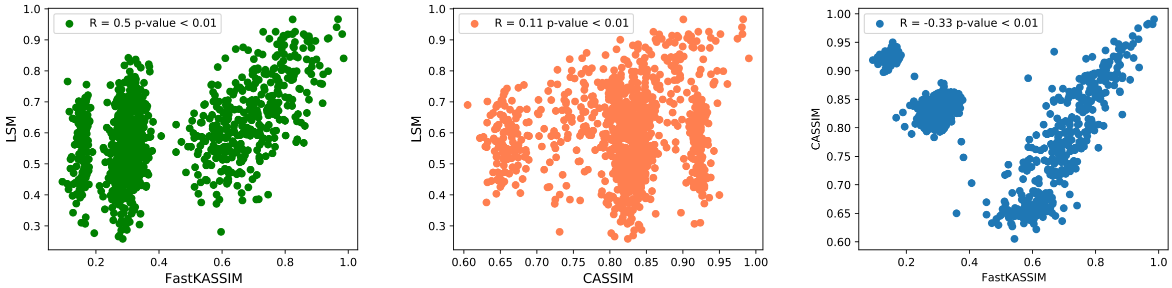
Figure 5 indicates that there are several regions of disagreement (vertical clustering). In one region of Figure 5a, FastKASSIM assigned low scores (less than 0.4) despite LSM ranging from 0.258 to 0.842. Recall that in this corpus, every utterance resulting from the same prompt was perceived as syntactically similar, and utterances from two different prompts were perceived as syntactically dissimilar. We find that in 248 out of 249 cases where FastKASSIM assigned a score below 0.4 yet LSM assigned a high score (above 0.6), the two documents being compared came from different prompts. This implies that FastKASSIM indeed is discriminating between different syntactic structures, whereas LSM may be picking up on similarities other than syntax, as expected.

Figure 5b does not indicate any obvious relationship between CASSIM and LSM. More surprisingly, Figure 5c shows that for document pairings that FastKASSIM deems syntactically dissimilar (values of less than 0.5), there is a very strong negative correlation between FastKASSIM and CASSIM ($R = -0.783$, $p < 0.001$). Visually, there is a cluster of pairings where FastKASSIM assigns a value less than 0.4 but CASSIM assigns a value larger than 0.75. We find that in 677 of the 678 pairings in this cluster, the documents originate from different prompts, indicating that the documents are syntactically dissimilar.

We evaluate these discrepancies in Table 2 in terms of each metric's ability to correctly identify documents originating from the same prompt as syntactically similar and those from differ-

[5] We use z-score standardization, $\frac{x-\mu}{\sigma}$.

[6] We compute LSM using an implementation publicly available at https://github.com/miserman/lingmatch.



(a) (b) (c)

Figure 5: (a). FastKASSIM v. LSM. Moderately strong positive correlation with $R = 0.5$ and $p < 0.01$. (b). CASSIM v. LSM. Weak but statistically significant positive correlation with $R = 0.11$ and $p < 0.01$. (c). FastKASSIM v. CASSIM. Moderately strong negative correlation with $R = -0.33$, $p < 0.01$.

ent prompts as syntatically dissimilar using accuracy, recall, and precision[7]. In addition to CASSIM and LSM, we include an evaluation of BERTScore (Zhang et al., 2019b) and Sentence-BERT (Reimers and Gurevych, 2019). While they are not syntax metrics, they are strong embedding-based metric which may account for syntactic forms.

As similarity scores lie between 0.0 and 1.0, we use 0.5 as a boundary between similar and dissimilar documents. We reason that in unknown contexts, 0.5 is neutral. We also compare against quantile transformations of each baseline metric, which map each metric's scores to a uniform distribution (note this is an unfair advantage, since in real-time deployment, one cannot observe the entire distribution of values), due to their apparent biases in Figure 5. In Table 2, we find that FastKASSIM holistically outperforms both CASSIM and LSM, with the exception of similar document recall and dissimilar document precision. In these cases, CASSIM achieves perfect precision and recall because it only classified one document pair as dissimilar. BERTScore similarly yields a small range in similarity scores on this corpus. After undergoing a quantile transformation, BERTScore's sensitivity to syntactic differences is magnified, and it performs well in the aforementioned categories but underperforms FastKASSIM in accuracy, similarity precision and dissimilarity recall.

As indicated by its low Dissimilar Document Recall in Table 2, we found CASSIM frequently assigned high similarity scores to syntactically different documents. This likely comes from the bias in its normalized Edit Distance as discussed in Section 3.1, and we show the significant improvements achieved by FastKASSIM.

Metric	Acc.	SR	SP	DR	DP
LSM	46.2	92.5	30.8	30.7	92.5
LSM$_a$	65.6	81.1	40.6	60.4	90.6
CASSIM	25.1	**100.**	25.0	0.11	**100.**
CASSIM$_a$	48.8	47.7	23.8	49.2	73.8
BERTScore	25.0	100.	25.0	00.0	00.0
BERTScore$_a$	74.6	99.3	49.6	66.4	99.6
Sentence-BERT	18.9	19.8	74.0	2.70	0.20
Sentence-BERT$_a$	34.3	9.50	19.2	59.3	39.3
FastKASSIM	**88.3**	96.1	**69.1**	**98.5**	85.6

Table 2: Evaluation of LSM, CASSIM, BERTScore, Sentence-BERT and FastKASSIM in terms of Accuracy (Acc.), Similar Document Recall (SR), Similar Document Precision (SP), Dissimilar Document Recall (DR), and Dissimilar Document Precision (DP). Metric$_a$ denotes adjusting to a uniform distribution by quantile transformation.

Overall, our results indicate that with respect to human intuition of syntax, FastKASSIM is more robust than CASSIM, LSM and BERTScore. In Appendix D, we visualize comparisons of several pairs of parse trees along with their FastKASSIM and CASSIM scores.

5 Applications

FastKASSIM is a more accurate and efficient syntactic similarity metric than the current state-of-the-art, opening the possibility for investigating new hypotheses in data-heavy fields with large corpora. Existing applications use syntax metrics for classification (e.g. Posadas-Duran et al. (2014)) as well as analytically to measure hypotheses (e.g. Kaster et al. (2021)). Here, we use syntax as a linguistic style indicator in authorship attribution, and measure syntactic similarity to study communication accommodation in persuasive arguments.

[7] Exact expressions provided in Appendix B.

5.1 Persuasiveness of Syntax Accommodation

Early work in communication accommodation theory found that matching communication styles can create a sense of familiarity, which improves social and conversational outcomes (e.g. Curhan and Pentland (2007); Giles (2016)). While most existing work has focused on hypotheses at the word-level (Tan et al., 2016), we hypothesize that CMV arguments that are more syntactically similar to opinions may be more persuasive as well.[8]

On CMV, users write an original post describing an opinion and allow "challengers" to present arguments attempting to change their opinion. Original posters (OP) indicate whether their opinions have been changed by assigning a "delta," which we can use an indicator of successful persuasion. We computed the syntactic similarity between a challenger's initial argument and an original poster's (OP) original opinion. This choice is made because the OP presents their full opinion in their original post, and a challenger typically presents their central argument in their initial challenge (Tan et al., 2016). While many CMV studies predict persuasion outcomes, prediction tasks do not reveal the actual bidirectional relationship between syntactic similarity and persuasion. We use FastKASSIM to analyze this relationship.

We find that arguments which eventually lead to deltas ($\mu = 0.307$) tend to be more syntactically similar to original opinions than unsuccessful ($\mu = 0.263$) arguments ($t = 19.016$; $p < 0.001$). However, this finding does not imply on its own that *syntactically similar arguments are more persuasive*. Thus, we also examined the converse by computing the persuasion rates (proportion of threads receiving deltas) of the most syntactically similar and dissimilar arguments. We computed the syntactic similarity of each pairing of initial arguments and original opinions and grouped them into the top and bottom 33% of syntactic similarity. This resulted in a minimum syntactic similarity value of 0.341 for the top 33% ($\mu = 0.453$) and maximum of 0.171 for the bottom 33% ($\mu = 0.096$). We found that threads grouped in the bottom 33% of syntactic similarity only had a persuasion rate of 6.377%, while the persuasion rate for threads grouped in the top 33% was nearly twice that, 12.347% ($t = 19.135$; $p < 0.001$). These findings support the hypothesis that similar syntactic patterns play a role in persuasion — may be an indication of stylistic familiarity for the OP.

Features	**Acc.**$_{(\sigma)}$	**F1**$_{(\sigma)}$
Majority Baseline	0.767	0.868
Bag of Words	$0.892_{(0.02)}$	$0.867_{(0.02)}$
Bag of Words + Syntax	$0.923_{(0.02)}$	$0.922_{(0.01)}$
RoBERTa	$0.939_{(0.01)}$	$0.935_{(0.00)}$
RoBERTa + Syntax	$\mathbf{0.945}_{(0.01)}$	$\mathbf{0.938}_{(0.01)}$

Table 3: **Judgment** test set results comparing accuracy and weighted F1 score between unigram counts and unigram counts augmented with syntactic features. Standard deviation (σ) given in subscripts.

5.2 Authorship Attribution

Much work has examined methods for attributing authorship based upon linguistic features (Juola, 2008; Raghavan et al., 2010; Seroussi et al., 2011b). The **Judgment dataset** (Seroussi et al., 2011a) contained English judgments delivered by judges on the Australian High Court from 1913 to 1975. We classified whether 924 judgments were written by Sir Edward McTiernan or Sir George Rich during non-overlapping time periods (Rich's judgments from 1913-1928 and McTiernan's from 1965-2971). We follow the experimental design and preprocessing steps in Seroussi et al. (2011b)[9].

To capture semantics, one setting used normalized Bag of Words with Support Vector Machines and the other used a state-of-the-art fine-tuned RoBERTa (Liu et al., 2019) model[10]. We augmented both semantic settings with a syntactic similarity feature vector — for each classification instance, we randomly sampled 25 posts from the training set and computed the FastKASSIM syntactic similarity between judgments written by Rich and McTiernan, respectively. The syntactic similarity features consisted of the minimum, maximum, mean, and standard deviation of these comparisons. We evaluated our classifier on a 10% withheld testing set[11].

Table 3 shows that adding syntactic features to both semantic models results in gains in both accuracy and weighted F1. This is even the case when using RoBERTa; we achieve the strongest performance using RoBERTa with a weighted sum between textual and syntactic features, fine-tuned

[8] Appendix C includes full details on CMV and preprocessing.

[9] All experiments were computed on one RTX A6000 GPU.

[10] Base RoBERTa (123M parameters). We set an initial learning rate of 2e-5 and a 0.01 weight decay.

[11] We used 4 seeds to sample our data.

using modules from the frameworks in Gu and Budhkar (2021); Wolf et al. (2020). Syntactic similarity with reference documents may provide a strong indicator of writing style.

6 Conclusion

We have introduced FastKASSIM, which has runtime improvements that scale significantly with document sizes and achieves better agreement with human perception of syntactic differences compared to standard syntax metrics. These improvements are possible due to our Label-based Tree Kernel, which has an improved asymptotic runtime complexity and a corrected normalization. FastKASSIM also allowed us to verify hypotheses regarding the importance of syntax both in authorship attribution and social dynamics such as persuasion. These findings motivate further applications of syntax.

Acknowledgements

Many thanks to Lillian Lee and Chenhao Tan for their feedback and collaboration on early versions of this work. We are very grateful to Reihane Boghrati for providing access to their Mechanical Turk dataset from their work on CASSIM. Thanks to Jack Hessel, Kun Qian, Yu Li, Qingyang Wu, Sky Wang, Mert Inan, and Zhiyang Xu for their helpful feedback and discussions.

Limitations

Our work relies on a couple assumptions. Our main corpus for evaluation is the crowdsourced and human-annotated dataset from Boghrati et al. (2018). As a result, our claim to better represent human perception of syntax relies on the assumption that their annotators correctly filter out responses which are not actually syntactically similar to each prompt. They had an acceptable Cohen's Kappa of 0.53. Additionally, we only use corpora that are in English. Future work should look towards applying our general approach to other languages.

In our evaluation of FastKASSIM against LSM and CASSIM, we also evaluate its ability to correctly identify statements created from the same prompt as similar and statements created from different prompts as dissimilar (Table 2). In this evaluation, we assume that 0.50 is an acceptable threshold for syntactic similarity and dissimilarity, because without any contextual information, one would assume that there are an equal amount of similar and dissimilar documents. Despite this, we still performed quantile adjustments for each comparison metric, uniformly distributing the scores between 0.0 and 1.0. This gives is an unfair advantage for the comparison metrics (i.e., CASSIM, LSM, and BERTScore), since "in-the-wild" it is impossible to obtain the eventual distribution of scores. Future work may consider methods to rebalance each of these scores, including conducting human evaluation to evaluate whether 0.50 is an acceptable threshold for syntactic similarity both before and after each metric undergoes an adjustment to the uniform distribution.

FastKASSIM is a metric for syntactic similarity between a pair of utterances or documents. However, similarity is only one dimension of syntax, which removes some granularity — for instance, syntactic similarity cannot explain which specific productions are shared. Similarity metrics like FastKASSIM instead afford opportunities in a variety of other applications, such as syntactic coherence in language generation and verifying computational social science hypotheses.

Additionally, parsing is still a significant bottleneck in runtime. Future work may wish to consider ways to mitigate the cost of parsing. One may also consider using sequential modeling to generate syntactic parse trees, or to directly model the output of FastKASSIM.

Ethical Considerations

Our study makes use of three datasets. First is the set of prompts collected in Boghrati et al. (2018), which involved anonymous participants creating fictional statements, so there is no personal information involved. Second is the publicly available r/ChangeMyView dataset collected by Tan et al. (2016), which consists of statements made by users behind typically anonymous aliases. Lastly is the publicly available WikiQA corpus (Yang et al., 2015), which does not contain identifying information.

In our r/ChangeMyView application studying the relationship between syntactic similarity and persuasion, we make the assumption that r/ChangeMyView is a community representative of online arguments. However, partially due to its anonymity, it is unknown whether r/ChangeMyView is a representative sample with diversity in location, educational background, socioeconomic status, ethnicity, and many other important factors. An ideal study should be able to control for proxies for individual traits in order to isolate the impact of syntax itself.

Generally, while most algorithms are not inherently unethical, there is often potential for abuse in their applications. The individual computations in the FastKASSIM algorithm do not have any negative implications, but it is possible to use syntactic similarity for unethical downstream tasks. For instance, because syntax is an important aspect of writing style, it is possible that users may try to adversarially uncover an anonymous author's identity. We do not condone the use of FastKASSIM for any unlawful or morally unjust activities. We do not propose any new tasks that would introduce unethical activity.

References

Noufa Abdulaziz Alnajran. 2019. *An integrated semantic-based framework for intelligent similarity measurement and clustering of microblogging posts*. Ph.D. thesis, Manchester Metropolitan University.

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.

Anthony L Baker, Sean M Fitzhugh, Lixiao Huang, Daniel E Forster, Angelique Scharine, Catherine Neubauer, Glenn Lematta, Shawaiz Bhatti, Craig J Johnson, Andrea Krausman, et al. 2021. Approaches for assessing communication in human-autonomy teams. *Human-Intelligent Systems Integration*, 3(2):99–128.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Natalia Bogdanova-Beglarian, Tatiana Sherstinova, Olga Blinova, and Gregory Martynenko. 2016. An exploratory study on sociolinguistic variation of russian everyday speech. In *International Conference on Speech and Computer*, pages 100–107. Springer.

Reihane Boghrati, Joe Hoover, Kate M Johnson, Justin Garten, and Morteza Dehghani. 2016. Syntax accommodation in social media conversations. In *CogSci*.

Reihane Boghrati, Joe Hoover, Kate M Johnson, Justin Garten, and Morteza Dehghani. 2018. Conversation level syntax similarity metric. *Behavior research methods*, 50(3):1055–1073.

Reihane Boghrati, Kate M Johnson, and Morteza Dehghani. 2017. Generalized representation of syntactic structures. In *CogSci*.

Adam Boncz. 2019. *Communication as Joint Action: The role of cognitive alignment and coupling*. Ph.D. thesis, Central European University.

Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. 1997. Syntactic clustering of the web. *Computer networks and ISDN systems*, 29(8-13):1157–1166.

Tanya L Chartrand, William W Maddux, and Jessica L Lakin. 2005. Beyond the perception-behavior link: The ubiquitous utility and motivational moderators of nonconscious mimicry. *The new unconscious*, pages 334–361.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 263–270.

Jared R Curhan and Alex Pentland. 2007. Thin slices of negotiation: Predicting outcomes from conversational dynamics within the first 5 minutes. *Journal of Applied Psychology*, 92(3):802.

Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. 2011. Mark my words! linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web*, pages 745–754.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Characterizing stylistic elements in syntactic structure. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1522–1533.

Howard Giles. 2016. Communication accommodation theory. *The international encyclopedia of communication theory and philosophy*, pages 1–7.

Ken Gu and Akshay Budhkar. 2021. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 69–73.

Patrick GT Healey, Matthew Purver, and Christine Howes. 2014. Divergence in dialogue. *PloS one*, 9(6):e98598.

Masahiro Hoshida, Masahiko Tamura, and Yugo Hayashi. 2017. Lexical entrainment toward conversational agents: An experimental study on top-down processing and bottom-up processing. In *Proceedings of the 5th International Conference on Human Agent Interaction*, HAI '17, page 189–194, New York, NY, USA. Association for Computing Machinery.

Molly E Ireland and James W Pennebaker. 2010. Language style matching in writing: synchrony in essays, correspondence, and poetry. *Journal of personality and social psychology*, 99(3):549.

Patrick Juola. 2008. *Authorship attribution*, volume 3. Now Publishers Inc.

Maurits Kaptein, Deonne Castaneda, Nicole Fernandez, and Clifford Nass. 2014. Extending the similarity-attraction effect: The effects of when-similarity in computer-mediated communication. *Journal of Computer-Mediated Communication*, 19(3):342–357.

Marvin Kaster, Wei Zhao, and Steffen Eger. 2021. Global explainability of bert-based evaluation metrics by disentangling along linguistic factors. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8912–8925.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430.

Lydia Kokkola and Ulla Rydström. 2022. Creativity and cognition in fiction by teenage learners of english. *Language and Literature*, 31(1):99–118.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Claire Little, David Mclean, Keeley Crockett, and Bruce Edmonds. 2020. A semantic and syntactic similarity measure for political tweets. *IEEE Access*, 8:154095–154113.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stephan Ludwig, Ko De Ruyter, Mike Friedman, Elisabeth C Brüggen, Martin Wetzels, and Gerard Pfann. 2013. More than words: The influence of affective content and linguistic style matches in online reviews on conversion rates. *Journal of Marketing*, 77(1):87–103.

William W Maddux, Elizabeth Mullen, and Adam D Galinsky. 2008. Chameleons bake bigger pies and take bigger pieces: Strategic behavioral mimicry facilitates negotiation outcomes. *Journal of Experimental Social Psychology*, 44(2):461–468.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Kate G Niederhoffer and James W Pennebaker. 2002. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21(4):337–360.

Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-leman in the bamboo: Novel amr graph metrics and a benchmark for amr graph similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Partha Pakray, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2011. Textual entailment using lexical and syntactic similarity. *International Journal of Artificial Intelligence and Applications*, 2(1):43–58.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Mateusz Pawlik and Nikolaus Augsten. 2011. Rted: a robust algorithm for the tree edit distance. *Proceedings of the VLDB Endowment*, 5(4):334–345.

AR Pereira and Nivio Ziviani. 2003. Syntactic similarity of web documents. In *Proceedings of the IEEE/LEOS 3rd International Conference on Numerical Simulation of Semiconductor Optoelectronic Devices (IEEE Cat. No. 03EX726)*, pages 194–200. IEEE.

Juan-Pablo Posadas-Duran, Grigori Sidorov, and Ildar Batyrshin. 2014. Complete syntactic n-grams as style markers for authorship attribution. In *Mexican International Conference on Artificial Intelligence*, pages 9–17. Springer.

Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 conference short papers*, pages 38–42.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* *and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

David Reitter and Johanna D Moore. 2007. Predicting success in dialogue. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*, pages 808–815. Association for Computational Linguistics.

David Reitter, Johanna D Moore, and Frank Keller. 2006. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *In Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

Kenji Sagae and Andrew Gordon. 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 192–201.

Ruchita Sarawgi, Kailash Gajulapalli, and Yejin Choi. 2011. Gender attribution: tracing stylometric evidence beyond topic and genre. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 78–86.

Yanir Seroussi, Russell Smyth, and Ingrid Zukerman. 2011a. Ghosts from the high court's past: Evidence from computational linguistics for dixon ghosting for mctiernan and rich. *University of New South Wales Law Journal, The*, 34(3):984–1005.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2011b. Authorship attribution with latent dirichlet allocation. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 181–189.

Svetlana Stoyanchev and Amanda Stent. 2009. Lexical and syntactic adaptation and their impact in deployed spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 189–192.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Esko Ukkonen. 1985. Algorithms for approximate string matching. *Information and control*, 64(1-3):100–118.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Christopher G Wetzel and Chester A Insko. 1982. The similarity-attraction relationship: Is there an ideal one? *Journal of Experimental Social Psychology*, 18(3):253–276.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019a. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Richong Zhang, Zhiyuan Hu, Hongyu Guo, and Yongyi Mao. 2018. Syntax encoding with application in authorship attribution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2742–2753, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Formalizing FastKASSIM

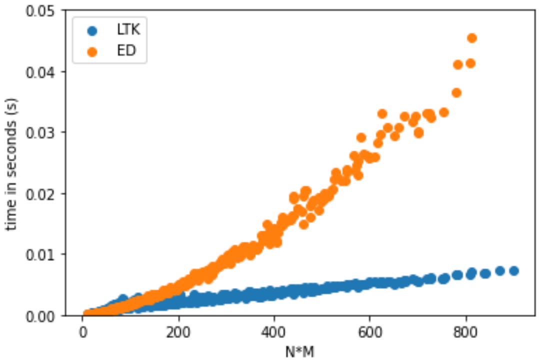


Figure 6: Runtime Comparison of Edit Distance (ED) and Label-based Tree Kernel (LTK) on WikiQA with varying NM (product of parse tree sizes).

The Label-based Tree Kernel recursively computes $\mathrm{Delta}_{\mathrm{lb}}$ across all n_1, n_2 pairs in parse trees P_1, P_2. The time complexity of $\Delta_{\mathrm{lb}}(n_1, n_2)$ has a ceiling of $O(L_1^h \times L_2^h)$, where L_i^h is the number of nodes at height h for a tree rooted at n_i, and h is the minimum height between the two trees. In the worst case scenario, Δ_{lb} of a fully uncached pair n_i, n_j results in recursive calls at every depth level. Δ_{lb} is computed for each node pair, so the ceiling of the tree kernel runtime is $O(NM)$, where N, M are the total number of nodes in P_1 and P_2, respectively.[12]

$O(NM)$ is a ceiling assuming the worst-case, where the labels are the same at each comparison, requiring full recursion. Let us consider there to be k shared labels in a pair of parse trees.

In each parse tree P_1, P_2, there will be $C_i^{(1)}, C_i^{(2)}$ connected components, one for each shared label $i \in [1, k]$, where a connected component consists of connected nodes with the same label. Out of the $C_i^{(1)}$ components in parse tree P_1, let $N_{i,j}^{(1)}$ be the size of each individual component j. So, for label $i = 1$, the number of comparisons follows:

$$O\left(\sum_{l=1}^{C_1^{(1)}} \sum_{m=1}^{C_1^{(2)}} N_{1,l}^{(1)} N_{1,m}^{(2)}\right)$$

which represents iterating through every pair of the $C_1^{(1)} \times C_1^{(2)}$ possible pairs of connected components and computing LTK. Then, for k shared labels, the worst-case runtime (i.e. the connected components do not form shared subtrees), we have equation 1:

$$O\left(\sum_{i=1}^{k} \sum_{l=1}^{C_i^{(1)}} \sum_{m=1}^{C_i^{(2)}} N_{i,l}^{(1)} N_{i,m}^{(2)}\right) = O\left(\sum_{i=1}^{k} N_i^{(1)} N_i^{(2)}\right) \tag{1}$$

where $N_i^{(1)}, N_i^{(2)}$ are the total number of nodes that have label i in P_1, P_2 respectively. However, recall from Algorithm 2 that LTK only iterates through pairs that share the same label; it does not matter if the connected components themselves are intertwined. Then, further simplifying this term we have equation 2:

$$O\left(\sum_{i=1}^{k} N_i^{(1)} N_i^{(2)}\right) = O(S_{12}) \tag{2}$$

where S_{12} is simply the total number of pairs in P_1, P_2 that have the same label. When all nodes have the same label, $S_{12} = NM$, consistent with the observed runtime ceiling. Empirically, we see that the *expectation* of S_{12} is much smaller than NM, as seen by the sublinear time scaling in Figure 6.

A.1 Scaling with Node Pairings (NM)

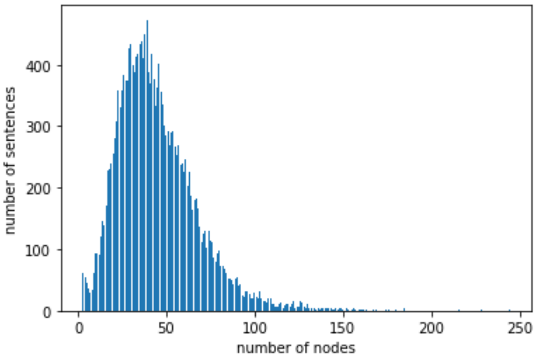


Figure 7: Statistics of Parse Trees in WikiQA

We first examine the relationship between LTK and NM, the number of possible node pairings, using the WikiQA corpus (Yang et al., 2015), a public dataset containing annotated question and answer pairs written in English. This dataset is chosen in order to compare FastKASSIM and CASSIM on well-structured text, and because of the ability to extract clean sentences of various length, which is crucial in determining NM. The experiment utilized 20,347 answer sentences from the WikiQA training set. We compute the parse trees prior to computing the runtimes of Edit Distance and LTK.

[12] Note that this is only possible due to Δ_{lb} caching the repetitive computations when iterating over node pairs.

Metric	Sim.	Dis.
LSM	70.4	56.0
LSM$_a$	72.2	42.6
CASSIM	82.1	82.0
CASSIM$_a$	48.3	50.6
BERTScore	89.9	85.1
BERTScore$_a$	85.7	38.0
Sentence-BERT	60.6	81.4
Sentence-BERT$_a$	26.7	57.7
FastKASSIM	73.1	31.7
FastKASSIM$_a$	86.1	38.0

Table 4: Average score assigned to similarity document pairings (Sim.) and dissimilar document pairings (Dis.) by each metric. Metric$_a$ denotes an adjustment to a uniform distribution using a quantile transformation.

Statistics of parse trees from the WikiQA-train corpus are shown in Figure 7.

As the corpus is rather large, we consider sentences with fewer than 30 nodes, which resulted in a total of 16,591,680 possible pairings. Then, pairings are grouped by the product of their nodes $N \times M$. For each value of $N \times M$, we track the cost of computing both the Edit Distance and LTK for all pairings if there are less than 10 pairs, or randomly sample 10 pairs if there are more. The average runtime for both Edit Distance and LTK for each value of $N \times M$ are shown in Figure 6.

B Metrics for Evaluating FastKASSIM, CASSIM, and LSM

In Section 4.3 and Table 2, we evaluated FastKASSIM, CASSIM, and LSM in terms of Similarity Accuracy, Similar Document Recall, Similar Document Precision, Dissimilar Document Recall, and Dissimilar Document Precision. These all follow the standard formulas for accuracy, recall, and precision. Adapted to our similarity context:

Similarity accuracy is the sum of the number of same prompt pairs receiving a score greater than 0.50 and the number of different prompt pairs receiving a score lower than 0.50 divided by the total number of pairings.

Similar document recall is the number same prompt pairs receiving a score greater than 0.50 divided by the total number of pairings originating from the same prompt.

Similar document precision is the number of same prompt pairs receiving a score greater than 0.50 divided by the total number of pairings receiving a score greater than 0.50.

Dissimilar document recall is the number different prompt pairs receiving a score less than 0.50 divided by the total number of pairings originating from different prompt.

Dissimilar document precision is the number of different prompt pairs receiving a score less than 0.50 divided by the total number of pairings receiving a score less than 0.50.

C Persuasiveness of Syntactic Similarity: Additional Context

C.1 CMV Background

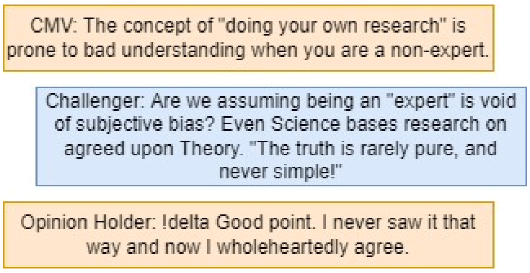
We investigate the role of syntax in persuasive arguments in the r/ChangeMyView[13] community (CMV) on Reddit. CMV users come in "good faith" that they are open to changing their view on a controversial topic. They write an original post describing an opinion and allow “challengers” to comment on their post and attempt to change their opinion. If their opinion is changed, the original poster (OP) will indicate this by assigning a “delta” (by typing either “!delta” or Δ in response to the persuasive comment). An OP may choose to present a rebuttal to a challenger, openly disagree with a challenger, or simply ignore a challenger (e.g., Figure 8). All Reddit users use anonymous aliases, unless they explicitly disclose their identity.

Earlier work found positive relationships between *behavioral mimicry* (mirroring behaviors) and in-person *negotiations* (Curhan and Pentland, 2007; Maddux et al., 2008). Yet, Healey et al. (2014) found that in general spoken conversations, peoples' syntactic patterns diverged from each other. We thus investigate the hypothesis that as a challenger on CMV continues to *engage* in an argument with an OP, their syntactic communication styles may begin to converge in order to “optimize for social differences.” Additionally, we hypothesize that challengers who utilize similar syntactic patterns, whether intentionally or not, may be more persuasive.

C.2 Dataset

We use the CMV dataset consisting of 18,363 posts and 1,114,533 comments written in English and collected by Tan et al. (2016). As in Tan et al. (2016), we examine discussion trees with at least 10 replies from challengers and at least one OP

[13] https://www.reddit.com/r/changemyview/



CMV: The concept of "doing your own research" is prone to bad understanding when you are a non-expert.

Challenger: Are we assuming being an "expert" is void of subjective bias? Even Science bases research on agreed upon Theory. "The truth is rarely pure, and never simple!"

Opinion Holder: !delta Good point. I never saw it that way and now I wholeheartedly agree.

Figure 8: An example of a thread on CMV. The OP presents a set of arguments defending their opinion (orange, top), inviting challengers to contest their opinion (blue). The OP acknowledged their opinion has been changed by assigning a delta (orange, bottom).

reply, in order to focus on discussions with "non-trivial" amounts of engagement. We also filter out posts which receive more than 10,000 comments in order to reduce noise from "outsiders" in cases where a post goes viral. When a challenger comments on an original post, it starts a "thread," with the original post (OP's opinion) taking on the root, index 0, and the challenger's comment taking on index 1. Each additional comment made in reply extends the thread. We are interested in syntactic accommodation, so we only consider threads that consist of conversations between the OP and a single challenger to eliminate confounders. These preprocessing steps results in a final dataset consisting of $15,986$ posts.

C.3 Dicusssion & Implications

In Section 5.1, we found two very statistically significant relationships between syntactic similarity and persuasive arguments. First, arguments that eventually lead to deltas tend to be more syntactically similar to original opinions compared to arguments that do not. Second, the arguments that are the most syntactically similar to original opinions actually were nearly twice as likely to receive deltas than the least syntactically similar arguments. Altogether, this may imply that syntactically similar arguments are more persuasive. This idea is supported by the rich body of work suggesting that similarity and communicative familiarity leads to improved social and conversational outcomes (Curhan and Pentland, 2007; Giles, 2016; Kaptein et al., 2014; Maddux et al., 2008; Wetzel and Insko, 1982).

D Parse Tree Examples

We visualize the constituency parse trees of several sentences taken from the corpus collected in Boghrati et al. (2018) using the online interface of the Berkeley Neural Parser[14], which uses the parser described in Kitaev et al. (2019).

We first compare the parse trees of the examples provided in Figure 1. Figure 9 compares the parse trees of the two similar documents shown in Figure 1. The first document is composed of one sentence — "I like swimming because it is cool." and the second document is also composed of one sentence — "I love running because it is fun." CASSIM assigned a score of 0.962, and FastKASSIM assigned a score of 0.928. The structure and composition of these two documents are nearly identical; the only difference is the production associated with the words "running" and "swimming."

Figure 10 is a visualization of the parse trees of the two dissimilar documents shown in Figure 1. The first document is composed of two sentences: "When we hate, we always move away from the grace of God. When we become resentful and unforgiving, the world around us seems spiteful and meaningless." The second document is composed of one sentence: "How can you be skiing if you are already swimming?" Beyond the differing number of sentences, the sentences in the first document individually appear structurally dissimilar compared to the sentence in the second document. FastKASSIM assigned a low score — 0.219, whereas CASSIM assigned a high score — 0.838.

Figure 11 compares the parse trees of the two single-sentence documents "How can you be skiing if you already swimming?" and "Knowledge is important to succeed." As is clear from Figure 11, the two sentences are structurally and compositionally quite different. FastKASSIM assigned a score of 0.439, whereas CASSIM assigned a score of 0.679.

Figure 12 compares the parse trees of two separate documents. The first document is composed of two sentences: "When we dream, we often search for deeper meaning. When we search for deeper meaning, other things become more nuanced too." The second document is composed of two sentences as well: "When we concentrate, we try to do better on a task. When we strive to do better, we end up doing better too." The structures of the two documents appear rather similar, but there do appear to be some differences in composition

[14] https://parser.kitaev.io/

(i.e., in terms of the constituent parts-of-speech). FastKASSIM assigned a score of 0.656, and CASSIM assigned a score of 0.837. While both scores are relatively high, FastKASSIM may be more penalizing towards these types of differences.

Figure 13 compares the parse trees of two separate documents. The first document is composed of four sentences: “I am old enough to draw freely upon my experience. Experience is more important than luck. Luck can turn. Experience lasts a lifetime.” The second document is composed of one sentence: “Being loving makes you become better.” Holistically, the structures of the two documents are quite different. Beyond the differing number of sentences in each document, there are also not any individual sentences between the two documents that appear particularly syntactically similar. FastKASSIM assigned a score of 0.15, whereas CASSIM assigned a score of 0.924.

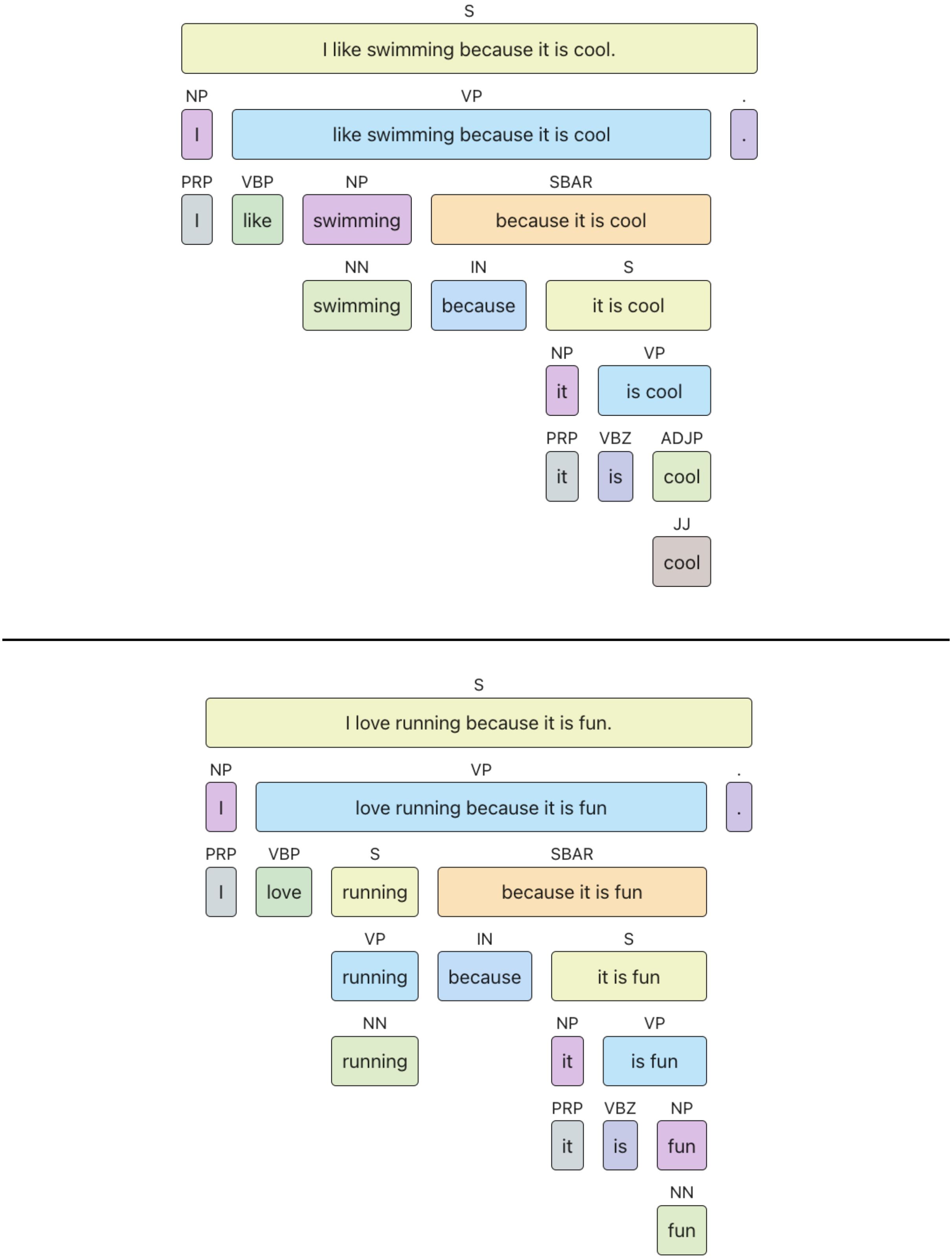


Figure 9: A comparison of the parse trees of two syntactically similar documents from Figure 1. Top document: "I like swimming because it is cool." Bottom document: "I love running because it is fun." FastKASSIM similarity score: 0.928; CASSIM similarity score: 0.962.

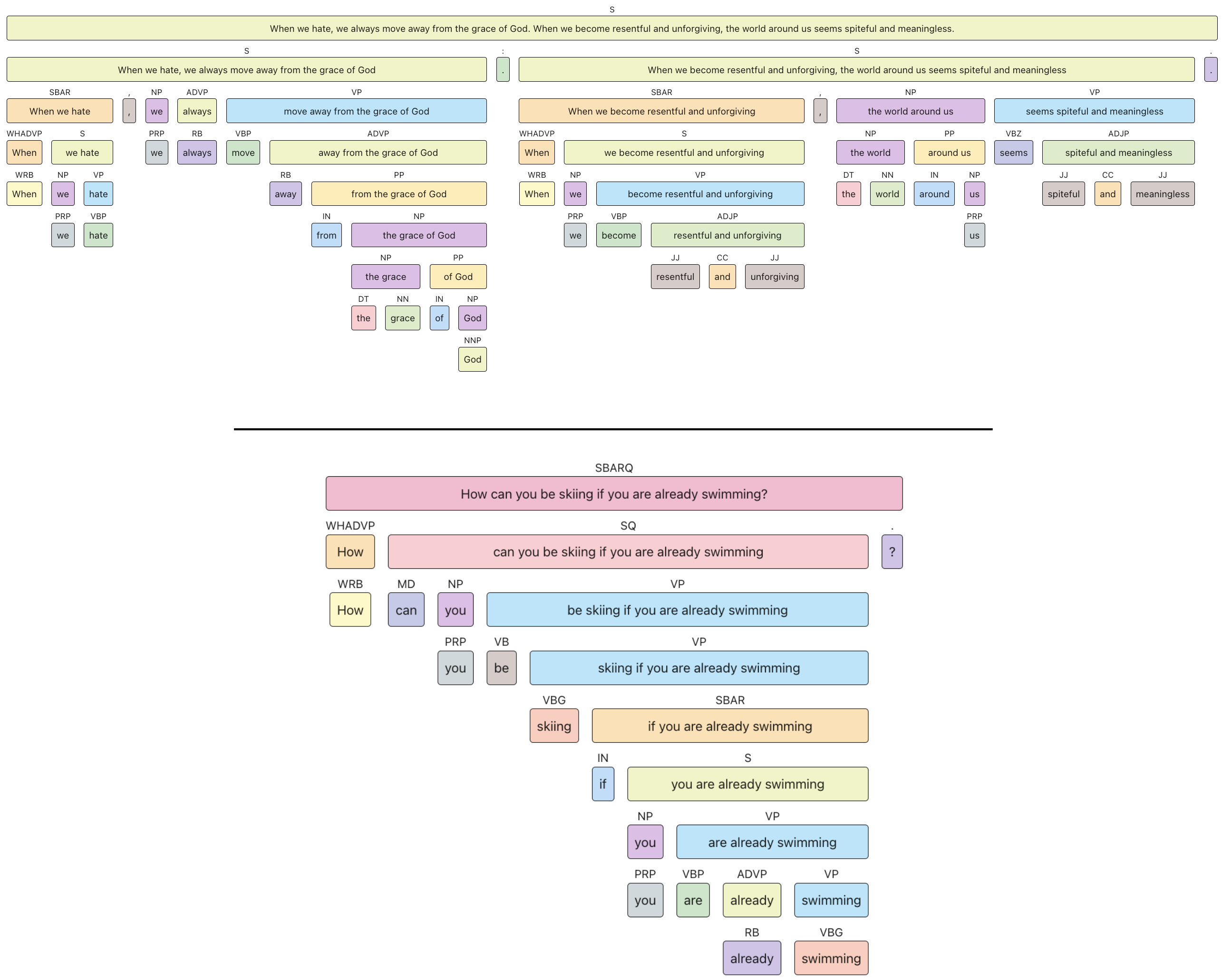

S
When we hate, we always move away from the grace of God. When we become resentful and unforgiving, the world around us seems spiteful and meaningless.
S
When we hate, we always move away from the grace of God
:
.
S
When we become resentful and unforgiving, the world around us seems spiteful and meaningless
.
.
SBAR
When we hate
,
,
NP
we
ADVP
always
VP
move away from the grace of God
SBAR
When we become resentful and unforgiving
,
,
NP
the world around us
VP
seems spiteful and meaningless
WHADVP
When
S
we hate
PRP
we
RB
always
VBP
move
ADVP
away from the grace of God
WHADVP
When
S
we become resentful and unforgiving
NP
the world
PP
around us
VBZ
seems
ADJP
spiteful and meaningless
WRB
When
NP
we
VP
hate
RB
away
PP
from the grace of God
WRB
When
NP
we
VP
become resentful and unforgiving
DT
the
NN
world
IN
around
NP
us
JJ
spiteful
CC
and
JJ
meaningless
PRP
we
VBP
hate
IN
from
NP
the grace of God
PRP
we
VBP
become
ADJP
resentful and unforgiving
PRP
us
NP
the grace
PP
of God
JJ
resentful
CC
and
JJ
unforgiving
DT
the
NN
grace
IN
of
NP
God
NNP
God
SBARQ
How can you be skiing if you are already swimming?
WHADVP
How
SQ
can you be skiing if you are already swimming
.
?
WRB
How
MD
can
NP
you
VP
be skiing if you are already swimming
PRP
you
VB
be
VP
skiing if you are already swimming
VBG
skiing
SBAR
if you are already swimming
IN
if
S
you are already swimming
NP
you
VP
are already swimming
PRP
you
VBP
are
ADVP
already
VP
swimming
RB
already
VBG
swimming

Figure 10: A comparison of the parse trees of two syntactically dissimilar documents. Top document: “When we hate, we always move away from the grace of God. When we become resentful and unforgiving, the world around us seems spiteful and meaningless.” Bottom document: “How can you be skiing if you are already swimming?” FastKASSIM similarity score: 0.219; CASSIM similarity score: 0.838.

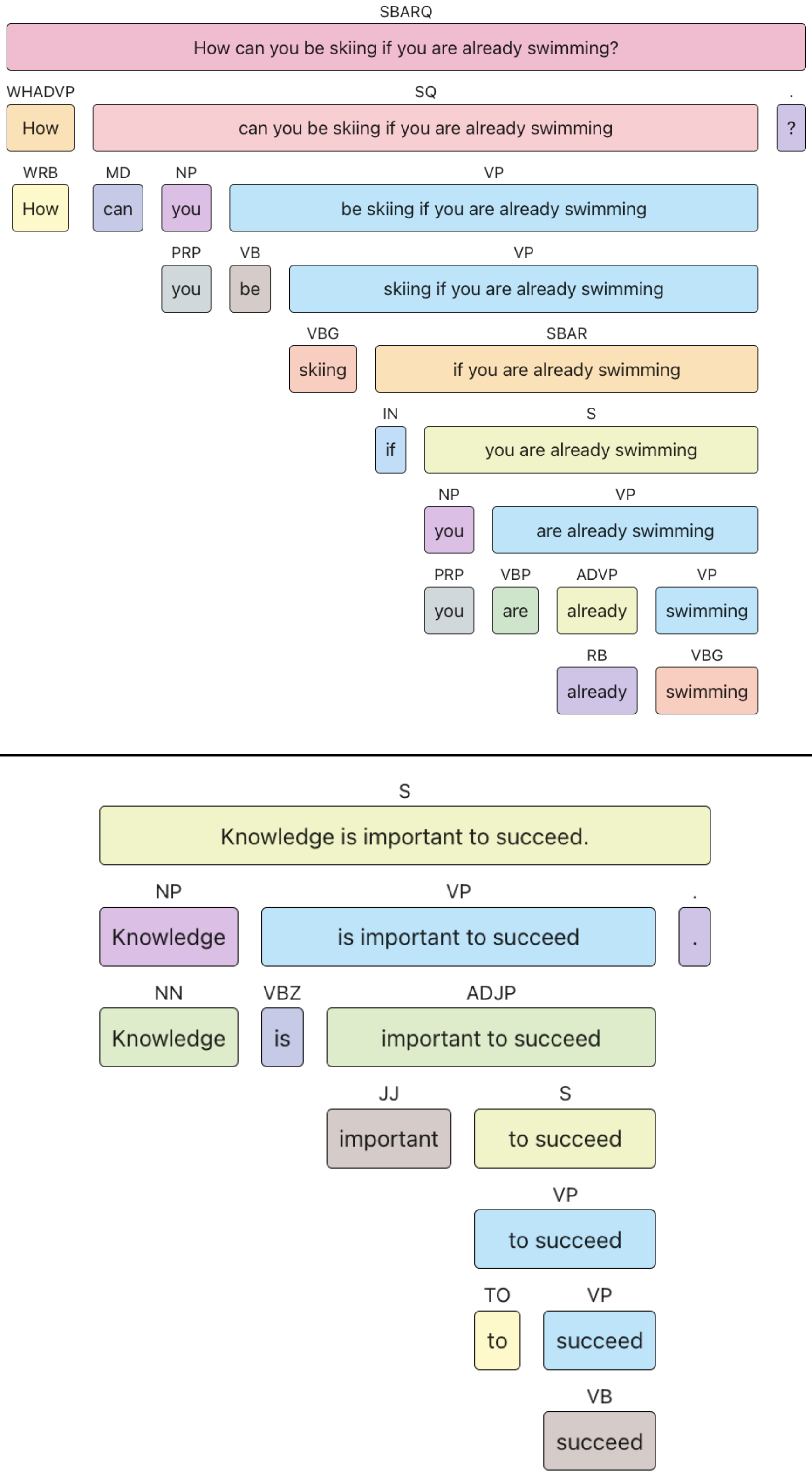


Figure 11: A comparison of the parse trees of two syntactically dissimilar documents. Top document: “How can you be skiing if you are already swimming?” Bottom document: “Knowledge is important to succeed.” FastKASSIM similarity score: 0.439; CASSIM similarity score: 0.679.

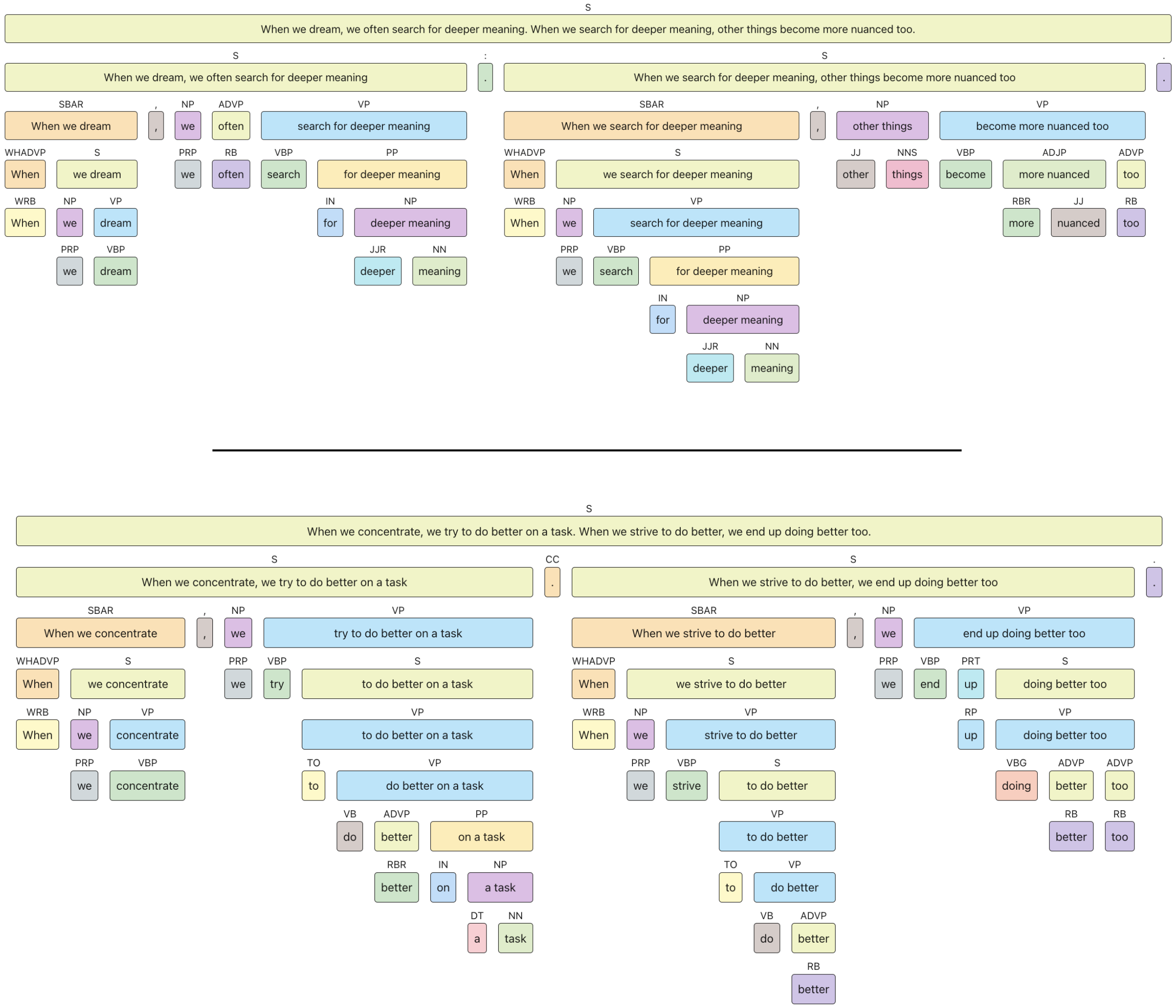


Figure 12: A comparison of the parse trees of two syntactically similar documents. Top document: “When we dream, we often search for deeper meaning. When we search for deeper meaning, other things become more nuanced too.” Bottom document: “When we concentrate, we try to do better on a task. When we strive to do better, we end up doing better too.” FastKASSIM similarity score: 0.656; CASSIM similarity score: 0.837.

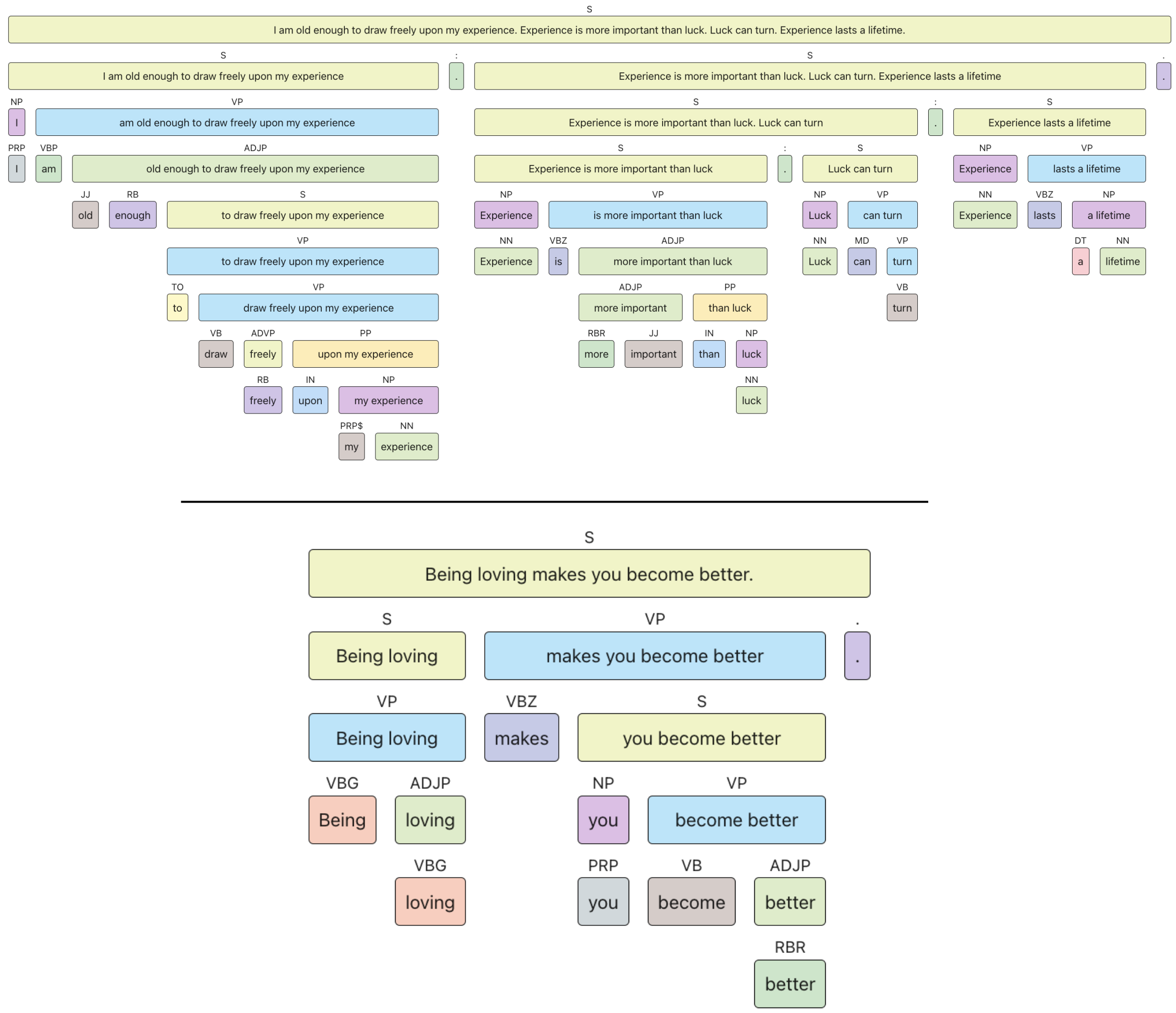

S
I am old enough to draw freely upon my experience. Experience is more important than luck. Luck can turn. Experience lasts a lifetime.
S
I am old enough to draw freely upon my experience
:
S
Experience is more important than luck. Luck can turn. Experience lasts a lifetime
.
NP
I
VP
am old enough to draw freely upon my experience
S
Experience is more important than luck. Luck can turn
:
S
Experience lasts a lifetime
PRP
I
VBP
am
ADJP
old enough to draw freely upon my experience
S
Experience is more important than luck
:
S
Luck can turn
NP
Experience
VP
lasts a lifetime
JJ
old
RB
enough
S
to draw freely upon my experience
NP
Experience
VP
is more important than luck
NP
Luck
VP
can turn
NN
Experience
VBZ
lasts
NP
a lifetime
VP
to draw freely upon my experience
NN
Experience
VBZ
is
ADJP
more important than luck
NN
Luck
MD
can
VP
turn
DT
a
NN
lifetime
TO
to
VP
draw freely upon my experience
ADJP
more important
PP
than luck
VB
turn
VB
draw
ADVP
freely
PP
upon my experience
RBR
more
JJ
important
IN
than
NP
luck
RB
freely
IN
upon
NP
my experience
NN
luck
PRP$
my
NN
experience
S
Being loving makes you become better.
S
Being loving
VP
makes you become better
.
VP
Being loving
VBZ
makes
S
you become better
VBG
Being
ADJP
loving
NP
you
VP
become better
VBG
loving
PRP
you
VB
become
ADJP
better
RBR
better

Figure 13: A comparison of the parse trees of two syntactically dissimilar documents. Top document: “I am old enough to draw freely upon my experience. Experience is more important than luck. Luck can turn. Experience lasts a lifetime.” Bottom document: “Being loving makes you become better.” FastKASSIM similarity score: 0.15; CASSIM similarity score: 0.924.