	Bandwidth	minLA
π_I (Standard)	17.64	82.39
π_R (Random)	20.94	294.43
π_b^* (Bandwidth)	6.75	101.23
π_m^* (minLA)	9.43	54.57

Table 1: Bandwidth and minimum linear arrangement scores for the specified permutation type averaged across 100000 Wikipedia sentences.

	Accuracy
π_I (Standard)	95.8
π_R (Random)	94.8
π_b^* (Bandwidth)	96.2
π_m^* (minLA)	**97.5**
AdaSent (Zhao et al., 2015)†	95.5
CNN+MCFA (Amplayo et al., 2018)†	94.8

Table 2: Accuracy on the **SUBJ** dataset using the specified ordering of pretrained representations for the finetuning LSTM. † indicates prior models that were evaluated using 10-fold cross validation instead of a held-out test set.

bandwidth cost. We also find the comparison of the standard and random orderings to be evidence that human orderings of words to form sentences (at least in English) are correlated with these objectives, as they are significantly better with respect to these objectives as compared to random orderings. Refer to Figure 3 for a larger example.

Downstream Performance In Table 2, we present the results on the downstream task. Despite the fact that the random permutation LSTM encoder cannot learn from the word order and implicitly is restrained to permutation-invariant features, the associated model performs comparably with previous state of the art systems, indicating the potency of current pretrained embeddings and specifically ELMo. When there is a deterministic ordering, we find that the standard ordering is the least helpful of the three orderings considered. We see a particularly significant spike in performance when using permutations that are minLA optimal and we conjecture that this may be because minLA permutations improve on both objectives on average and empirically we observe they better maintain the order of the original sentence (as can be seen in Figure 1).

5 Related Work

This work draws upon inspiration from the literature on psycholinguistics and cognitive science. Specifically, dependency lengths and the existing minimization in natural language has been studied under the dependency length minimization (DLM) hypothesis (Liu, 2008) which posits a bias in human languages towards constructions with shorter dependency lengths.[3]

In particular, the relationship described between random and natural language orderings of words to form sentences as in Table 1 has been studied more broadly across 37 natural languages in Futrell et al. (2015). This work, alongside Gildea and Temperley (2010); Liu et al. (2017); Futrell et al. (2017) helps to validate the extent and pervasiveness of DLM in natural languages. More generally, this literature body has proposed a number of reasons for this behavior, many of which center around the related notions of efficiency (Gibson et al., 2019) and memory constraints (Gulordava et al., 2015) for humans. Recent research at the intersection of psycholinguistics and NLP that has tried to probe for dependency-oriented understanding in neural networks (primarily RNNs) does indicate relationships with specific dependency-types and RNN understanding. This includes research considering specific dependency types (Wilcox et al., 2018, 2019a), word-order effects (Futrell and Levy, 2019), and structural supervision (Wilcox et al., 2019b).

Prompted by this, the permutations considered in this work can alternatively be seen as linearizations (Langkilde and Knight, 1998; Filippova and Strube, 2009; Futrell and Gibson, 2015; Puzikov and Gurevych, 2018) of a dependency parse in a minimal fashion which is closely related to Gildea and Temperley (2007); Temperley and Gildea (2018). While such linearizations have not been well-studied for downstream impacts, the usage of dependency lengths as a constraint has been studied for dependency parsing itself. Towards this end, Eisner and Smith (2010) showed that using dependency length can be a powerful heuristic tool in dependency parsing (by either enforcing a strict preference or favoring a soft preference for shorter dependencies).

[3] In this work, we partially deviate from this linguistic terminology, which primarily deals with the measure defined in Equation 2, and prefer algorithmic terminology to accommodate the measure defined in Equation 1 and disambiguate these related measures more clearly.