Multi-Task Learning for Contextualized Word Embeddings

Bjarne de Jong
10657894

Tycho Grouwstra
6195180

Silvan de Boer
12158054

Abstract

A hierarchical, multi-task network is trained on supervised data to develop contextualized word embeddings that are specifically good at disambiguation of words based on context. Word embeddings from a model trained on sequential metaphor labelling and natural language inference are shown to significantly outperform the unsupervised baselines. Analysis of the model suggests that the metaphor task helps to distinguish specifically between two word senses of which one is metaphorical.

1 Introduction

Word embeddings are an important means to capture lexical semantics, and as such relevant in fields such as information retrieval and natural language processing. With their fixed dimensionality, they lend themselves well as features for deep learning models. Traditional word embeddings (Mikolov et al., 2013; Pennington et al., 2014) are limited, because they are independent of context and as such only represent a general word meaning in a corpus.

Recently, unsupervised contextualized word embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) were introduced. Almost any model can be improved by using these context-dependent embeddings. The Word-in-Context (WiC) task was developed to measure how well a word embedding captures context-dependent meaning variation (Pilehvar and Camacho-Collados, 2018). Despite the success of unsupervised embeddings of ELMo and BERT, this task shows that there is room for improvement.

In this work, we investigate whether supervised data can be leveraged to better capture this context-dependent meaning variation compared to a purely unsupervised baseline. Inspired by successes in multi-task learning (e.g. Hashimoto et al., 2016), we propose a hierarchical model that is trained on tasks of increasing complexity.

Metaphors are ubiquitous in language, and pose a challenge for general language understanding. Applications could benefit significantly from accurate models of metaphor (Shutova, 2015). Such applications include machine translation (metaphors are culture-specific), opinion mining (metaphor usage is influenced by emotion), and information retrieval (metaphors are ambiguous). Moreover, the metaphoricity of a word is fully determined by its context. We hypothesize that the sequential metaphor labelling task is specifically helpful in our multi-task model.

Additionally, we experiment with the part-of-speech tagging task (PoS) and the natural language inference task (NLI). In an extensive ablation study, the performance of different set-ups is investigated. Moreover, we compare word embeddings extracted from different parts of the model.

2 Related Work

2.1 Multi-Task Learning

Multitask learning (MTL) (Caruana, 1997) is an approach used in natural language processing (NLP) to increase generalization beyond training data, by training on various related tasks simultaneously while using a shared representation. The intuition here is that various tasks in NLP each cover one part of the over-arching structure and meaning in language, and that, as such, we are likely to improve performance on the more difficult tasks (including hyperlink detection, multi-word-expression detection) by bringing in some of what we have learned on the simpler tasks (including Part of Speech (PoS) tagging, chunking, logical type tagging, super-sense tagging, and semantic frames). (Bingel and Søgaard, 2017) Perform-

ing multitask learning in such a hierarchical setup involving a sequence of tasks of increasing complexity is called *Hierarchical Multi-Task Learning* (Hashimoto et al., 2016).

2.2 Word Embeddings

Word embeddings are generally learned in unsupervised fashion. Mikolov et al. (2013) introduces the skip gram model (word2vec) for learning distributional representations for a word given its context. In this model a word obtains a representation which should portray the general meaning of the word irrespective of its context. Pennington et al. (2014) improve on the word2vec model by using matrix factorization instead of the skip-gram arguing that their method can better capture global statistics and repetitive patterns in the training corpus.

2.3 Contextual Word Embeddings

ELMo (Embeddings from Languange Models (Peters et al., 2018)) introduces a powerful new manner of representing words whereas previous work uses a fixed representation for a word regardless of its context. ELMo models the representation of a word using its context, so that the same word ocurring in different contexts will have a different representation, unlike in the GloVe model. The ELMo model consists of three different parts which all capture a word representation at different levels of context. The first level – a character CNN – only captures word-level contexts and therefore should behave roughly similar to models such as GloVe and Word2vec. The second level builds a more syntactical representation of word meaning according to the authors. The third and final level captures a semantic representation based on the context of the word according to the authors.

Further improvement in the field of contextual word embeddings were made by Devlin et al. (2018). They replaced the LSTM layers of the ELMo model with transformers layers, made the model much deeper from two bi-directional LSTM layers up to twelve layers of transformers, trained on more data, and introduced an innovative new training method. BERT obtains a relative performance increase when compared to ELMo ranging from 5% up as much as 66% in various tasks according to Devlin et al. (2018). The current state-of-the-art performance on our evaluation task WiC is also attributed to a BERT model, called BERT-large (Wang et al., 2019)[1].

3 Methods

3.1 Tasks

We have identified three tasks that could be useful for the WiC task. From simple to increasingly difficult these are Part-of-Speech tagging (POS), Metaphor Detection in Context (MET) and the Natural Language Inference (NLI) task.

Each subsequent task conceptually requires more semantic understanding and should therefore improve the performance on WiC. Furthermore starting with simpler tasks and training on increasingly more difficult task in a curriculum learning fashion has been shown to be beneficial to individual tasks by Hashimoto et al. (2016).

3.2 Model Architecture

To investigate the influence of joint multi-task learning on the effectiveness of using contextualized word embeddings to capture context-dependent lexical meaning variation, we train a network similar to that defined by Hashimoto et al. (2016). The network is displayed in Figure 1.

The network is organized hierarchically, consisting of a bi-directional LSTM for each consecutive task. The order of tasks is based on complexity: POS, MET and then NLI. This is inspired by Hashimoto et al. (2016), who use the same approach.

Each LSTM has a hidden size of 200 units (100 for each direction). These hidden states will serve as our contextualized word embeddings. The POS and MET task are based on word-level classification. Therefore, the LSTMs of the POS and MET are connected to a linear layer serving as a classifier. NLI is sentence classification. To perform this task, we use the max-pooling architecture of Conneau et al. (2017). This network forms a sentence representation by max-pooling the word representations. Sentence pairs are joined and classified by a two-layer linear network. All tasks are modeled as multi-class classification problems and therefore use a softmax to output the final predictions.

3.3 Input word embedding

The words of the input sentence are embedded as a 1,324-dimensional vector, which consists of the

[1]The leaderboard of the task can be found at https://pilehvar.github.io/wic/

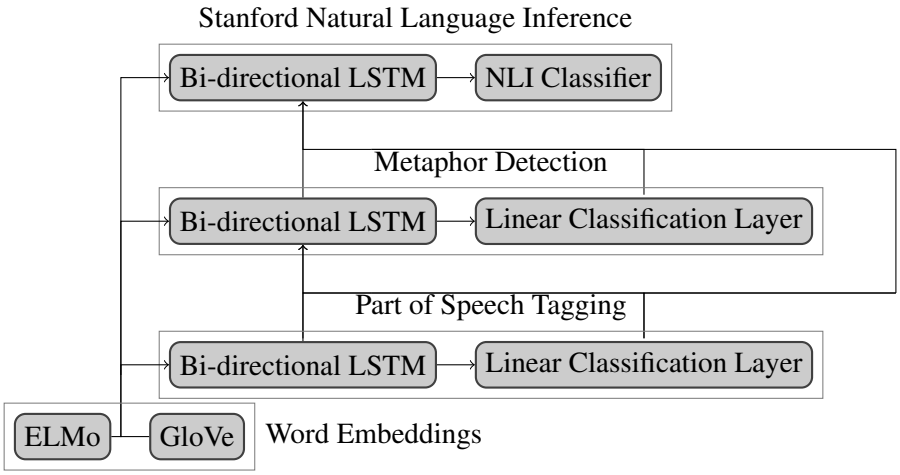

Figure 1: The hierarchical network architecture used. The model consists of three LSTM layers one for each model. Each LSTM layer receives as input the concatenation of the outputs of all lower tasks and the word-level embeddings from GloVe and ELMo.

concatenation of:

- pre-trained GloVe embeddings (840B, 300 dimensions) [2]
- pre-trained ELMo embeddings (5.5B) [3]. These embeddings are trained on a dataset of 5.5B tokens consisting of Wikipedia (1.9B) and all of the monolingual news crawl data from WMT 2008-2012 (3.6B). Model weights are fixed, mixing weights are fixed to average the three layers. As is generally suggested, prior to usage we would first run some batches to 'warm up' our ELMo embeddings [4], to ensure results are more reproducible and constant.

We base this embedding on the work of Gao et al. (2018), who perform only the MET task with this input embedding.

The input of all three LSTM layers consists of this concatenation of word level embeddings generated by GloVe and ELMo, along with the hidden state and classifier outputs of the lower levels. This means for example that the NLI layer receives the input embedding, the hidden state and classifier output of the MET layer, as well as the hidden state and classifier output of the POS layer.

[2] https://nlp.stanford.edu/projects/glove/

[3] https://allennlp.org/elmo

[4] https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md#notes-on-statefulness-and-non-determinism

3.4 Training

We trained the same hierarchical architecture on every subset of tasks. For each subset of tasks the architecture of the model was not changed to isolate performance difference that may arise due to changes in the model architecture (e.g. capacity). Moreover, Hashimoto et al. (2016) have shown that using the full model when training a subset of the tasks has a slight improvement over training a model with the unnecessary layers removed.

Each model was trained for 20 epochs using a separate SGD optimizer for each task. The hyperparameters used follow those from Hashimoto et al. (2016), where successive regularization is implemented by scaling the learning rate of lower-level tasks when training on higher-level tasks. If a lower-level task is left out, then a higher-level task will not perform successive regularization on those layers of the model. A batch size of 64 sentences was used for the word-level tasks, and 64 sentence pairs were used for the NLI task.

For the two word-level tasks, the loss was calculated on a per-word basis, while for the NLI task the loss was calculated for the two sentence pairs. Care was taken to alleviate the problem of class imbalance in the MET task, by sampling from the non-metaphor labels until the number of occurrences roughly matched that of the metaphor words. Furthermore, a masked version of the word-level cross-entropy loss was used to ignore the labels that were not sampled.

4 Experiments

4.1 Data

For each of the three tasks a dataset was acquired.

4.1.1 Part of Speech Tagging

For the Part of Speech Tagging tasks we considered two different datasets. The first one is the Wall Street Journal (WSJ) corpus from the Penn Treebank. The reason we considered using this is that it was also used in Hashimoto et al. (2016). Secondly, we considered the use of the VUA corpus used in Gao et al. (2018), as the domain of the data more closely aligns with that of the rest of the tasks. In the end we opted to use the VUA dataset, as the performance we reached using it usually exceeded that of the WSJ corpus. The VUA corpus is annotated on a word level and contains 17 different Part-of-Speech tags, and it contains roughly 6,300 sentences in the training set, roughly 1,500 sentences in the validation set, and roughly 2,700 sentences in the test set, following the split used in Gao et al. (2018).

4.1.2 Sequential Metaphor Detection

For the Sequential Metaphor Detection task the VUA corpus was also used following the work of Gao et al. (2018). It contains an annotation for each word in a sentence whether or not the word is used in a metaphorical sense. Due to the nature of the occurrence of using metaphorical word senses in a sentence, there is large class imbalance, where roughly 90% of the annotated words in the training set are not used in a metaphorical sense. We used the same split here as for the Part of Speech Tagging task.

4.2 Natural Language Inference

For the Natural Language Inference tasks we used the SNLI dataset (Bowman et al., 2015). The dataset consists of sentence pairs annotated with labels classifying the pairs into one of three categories, corresponding to the following relationships:

- the second sentence *contradicts* the first sentence;
- the second sentence logically *follows* from the first sentence (*entailment*);
- the second sentence and first sentence are *neutral* (elaborations and such).

The training set, validation set, and test set consist of 550 thousand, 10 thousand, and 10 thousand sentence pairs, respectively.

4.3 Word-in-Context evaluation

On the Word-in-Context (WiC) task we use accuracy as our evaluation metric. The WiC dataset of Pilehvar and Camacho-Collados (2018) is specifically designed to test context-sensitive word representations. Every entity in the dataset consists of a target word (verb or noun), and two contexts. The task is to classify whether the target word does or does not have the same meaning in both contexts. In this dataset, context is essential, because the target word is the same in both contexts.

For this task, we produce contextualized embeddings for the word in both sentences, then evaluate their similarity in the embedding space. Pilehvar and Camacho-Collados (2018) evaluate two different methods of performing this classification: comparing the distance between the embeddings using cosine similarity, then judging whether this cosine similarity exceeds a given threshold, or using a Multi-Layer Perceptron (MLP). Using the cosine similarity approach, one would judge the two words as having been used in the same sense across the two sentences.

As the MLP approach has been demonstrated to yield inferior results as per Pilehvar and Camacho-Collados (2018), we have chosen to instead perform the consine similarity approach as well as a Support Vector Machine (SVM). However, as the SVM yielded us results inferior to that of the cosine similarity + threshold, we ended up settling for that instead. We determine our optimal threshold using a grid search, with intervals of 0.02 from 0.0 to 1.0, following Pilehvar and Camacho-Collados (2018).

Unfortunately, labels of the WiC test set have not been made public as of the time of writing. We have determined the threshold using a subset of the training set of equal size as the development set (638 sentence pairs), then tested these on the development set. As a result, our results may differ somewhat from those reported in Pilehvar and Camacho-Collados (2018).

To verify the significance of our results, we have chosen to use McNemar's test (McNemar, 1947), a paired non-parametric statistical hypothesis test for use on paired nominal data that has relatively low Type-I error (Dietterich, 1998). This tests the

Model	WiC accuracy
$ELMo_{012}$	57.5%
$ELMo_{012}$+GloVe	56.6%
Random	58.6%
NLI	61.9%
MET-NLI	**63.2%**
POS-MET-NLI	61.0%
BERT-large[5]	**68.4%**
Human[6]	80.0%

Table 1: WiC accuracy of the best hierarchical model MET-NLI compared to NLI, POS-MET-NLI, baselines, state-of-the-art and human performance.

null hypothesis of marginal homogeneity, which states that the two marginal probabilities for each outcome are the same.

4.4 Baselines

ELMo embeddings, pre-trained by Pilehvar and Camacho-Collados (2018), are used as a baseline model. These word embeddings are dependent on context, but the ELMo model is derived from unsupervised data only. Therefore, the model lends itself for investigating the effect of adding supervised data in our hierarchical model. We test the WiC performance of different layers of ELMo:

- $ELMo_0$: take the character-based word embedding as our word embedding (should be random, because it is independent of context);
- $ELMo_1$: take the first LSTM hidden layer as word embedding;
- $ELMo_2$: take the second LSTM hidden layer as word embedding;
- $ELMo_{012}$: take the average of the above three embeddings.

Because the $ELMo_{012}$ embedding is used in the input to our model, it is chosen as the main baseline. Moreover, we noticed that a random initialization of the hierarchical model performs better than ELMo. We choose this model as a second baseline. Lastly, we compare to the full input of the model ($ELMo_{012}$+GloVe), since it indicates the contribution of the model as well.

An executive or judical *office*. During his first year in ***office***.
It was an ***apology*** for a meal. The *Apology* of Socrates.
Fuel aircraft, ships, and cars. ***Fuel*** the debate on creationism.

Table 2: Example sentence pairs from WiC with one metaphorical word sense (bold) and one different sense (italics). These pairs are correctly classified by MET-NLI and incorrectly by NLI.

5 Results and Analysis

The WiC accuracies of the different embeddings of the baselines and hierarchical models are shown in Figure 2. Table 1 provides a summary.

The input of the hierarchical models - a concatenation of $ELMo_{012}$ and GloVe - performs worse than ELMo on its own. GloVe vectors are independent of context, and therefore disrupt the WiC classification based on cosine similarity. Interestingly, some layers of the model with random parameters perform better than the ELMo baseline. A random function of a word's context is already a better indication of semantic similarity than the bare ELMo and GloVe vectors. This indicates that there is a lot to gain in this task, and is a simple but promising result.

The model that performed best was trained on sequential metaphor labeling and natural language inference (MET-NLI). The hidden state of the metaphor layer performed best as a word embedding. McNemar's test shows that the model is a significant improvement over the baselines $ELMo_{012}$ ($p = 0.003$), $ELMo_{012}$+GloVe ($p = 0.000$) and random ($p = 0.013$).

The part-of-speech task seems to have a negative effect on every model. To investigate whether inappropriate data is the cause, two PoS data sets were compared. Models trained on the VUA PoS data perform on average 1.2% (absolute) better than those trained on the Penn Treebank PoS data. Sentences in the Penn Treebank come from news articles from the Wall Street Journal, which is a rather specific domain. This may induce a bias that causes the performance to drop. However, PoS seems to be a bad influence on the WiC performance regardless of the data set.

In accordance to our hypothesis, the metaphor

[5]Current state-of-the-art (Wang et al., 2019)

[6]Estimated by Pilehvar and Camacho-Collados, 2018

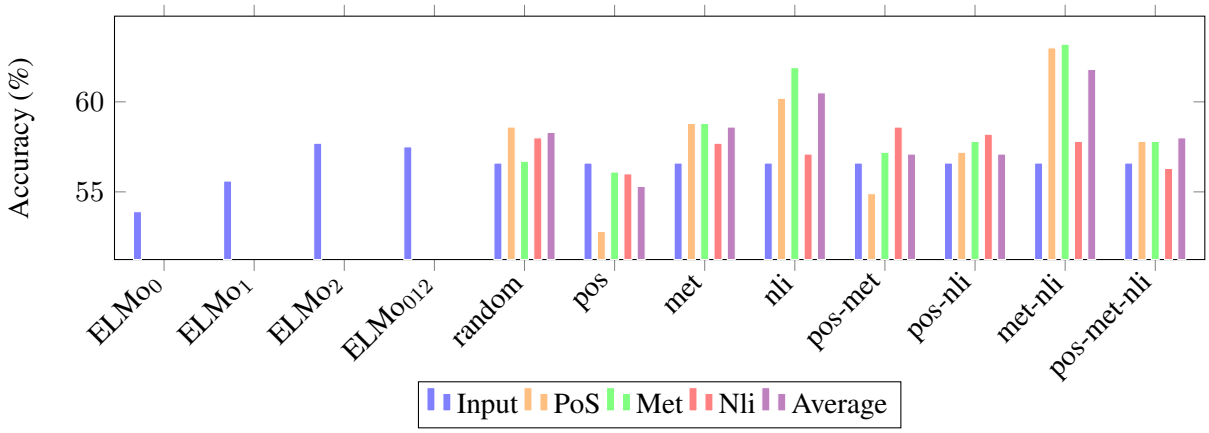

Figure 2: Accuracy on the Word-in-Context task. The ELMo baseline models are shown on the left. Results of the proposed hierarchical model are shown on the right, where the label indicates the tasks that were used in training. The 'random' model has the same architecture, with randomly initialised weights. The colors indicate from where the embedding is extracted: from the input or one of the hidden layers of the LSTMs.

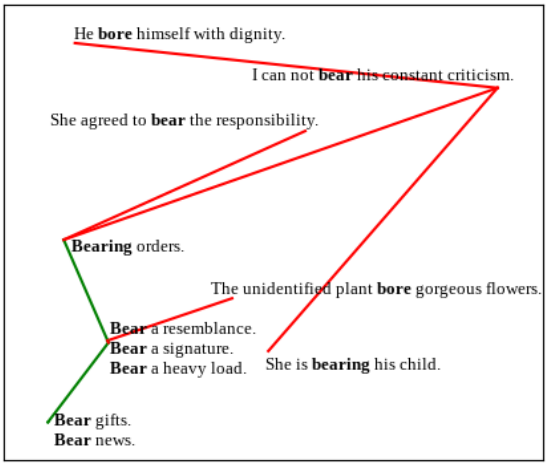

Figure 3: t-SNE representation of all word embeddings of the verb 'bear' in the training data of the WiC dataset. Green lines indicate a labelled sentence pair with the same meaning, red lines a pair with a different meaning.

labeling task seems to have a positive effect on the embedding performance. We suspect that metaphoricity provides a first general distinction between word senses, and aids disambiguation as such. We qualitatively compare the best model MET-NLI with the NLI model to investigate the influence of the metaphor labeling task [7]. The MET-NLI model, making use of data on metaphoricity, seems better able to classify sentence pairs where one word is metaphorical and the other is not. Examples of these cases are shown in table 2. This further supports our hypothesis, and could be subject to further research.

The MET-NLI embeddings of the verb 'bear' in the WiC training data are represented in figure 3. Word senses that are the same are connected by green lines. The words in the bottom left have the same sense and are relatively close to each other, indicating that the model is able to distinguish between senses for this word. Interestingly, it looks like the inclusion of a determiner in the sentence can have more effect than changing the object of the verb.

We have also sorted the false positives and false negatives based on the model's confidence score of the word senses matching, such as to analyze its worst mistakes. While we will defer a more complete overview to the appendix, false positives included largely reasonable mistakes, e.g. "He lived in *exile*" vs "She lived as an *exile*", some legitimate mistakes, e.g. "*Scallop* the hem of the dress"

[7] The metaphor layer embedding is used from both models to make the comparison fair. Note that it performs best on WiC in both models.

vs "*Scallop* the meat", as well as some that we feel might be issues with the dataset, e.g. "He could not conceal his *hostility*" vs "He could no longer contain his *hostility*". Do note however that granularity of word senses may seem somewhat subjective, and as such, human performance on this task only reaches 80.0% accuracy as well.

False negatives on the other hand largely appeared to be mistakes that would have seemed somewhat easier for a human, such as "To *command* an army or a ship" vs "*Command* the military forces". However, this section also contained some entries that would seem questionable to a human as well, e.g. "*Hail* a cab" vs "He was *hailed* as a hero".

6 Conclusion

We presented a hierarchical, multi-task supervised learning setup to train contextualized word embeddings that are specifically good at distinguishing senses of the same word in a different context. We show that the model trained on sequential metaphor labeling and natural language inference yields word embeddings that significantly outperform unsupervised baselines. In particular, our findings suggest that the metaphor labeling task help the embedding to better distinguish between word senses that differ in metaphoricity.

Further research could investigate the same hierarchical setup with different tasks, such as word sense disambiguation. Moreover, given the marginal improvements due to the metaphor task, it is worthwhile to continue investigating the usefulness of metaphor in disambiguation, and look for ways to quantify this relation.

References

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. *arXiv preprint arXiv:1808.09653*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple NLP tasks. *CoRR*, abs/1611.01587.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: 10,000 example pairs for evaluating context-sensitive representations. *arXiv preprint arXiv:1808.09121*.

Ekaterina Shutova. 2015. Design and evaluation of metaphor processing systems. *Computational Linguistics*, 41(4):579–623.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

A Statistical significance

Figure 3 shows the detailed results of our statistical significance test using McNemar's test. It demonstrates statistically significant improvement of results of our vua-snli:vua model over any of the benchmark models we used, showing that our multi-task learning setup has contributed to prediction accuracy on the WiC task.

	elmo012:input	elmo2:input	random:pos	vua-snli:input	vua-snli:vua
elmo012:input	NaN	1.000	0.636	0.624	0.003
elmo2:input	1.000	NaN	0.729	0.628	0.007
random:pos	0.636	0.729	NaN	0.208	0.013
vua-snli:input	0.624	0.628	0.208	NaN	0.000
vua-snli:vua	0.003	0.007	0.013	0.000	NaN

Table 3: p-values among our models using McNemar's test

B Top false positives and false negatives

This section contains the top false positives and false negatives as predicted by our best-performing and baseline models. Note that the final periods are preceded by a space to indicate that punctuation is considered as separate word tokens by our model. While some of these false positives/negatives might indicate errors in the dataset itself, the remaining entries nevertheless give us a better picture of the types of errors made.

[b]

cosine	sentence A	sentence B
0.967	He could not conceal his *hostility* .	He could no longer contain his *hostility* .
0.913	He lived in *exile* .	She lived as an *exile* .
0.906	*Engrave* a letter .	*Engrave* a pen .
0.902	*Indent* the documents .	*Indent* the paragraphs of a letter .
0.895	I could just make out her face in the *twilight* .	He loved the *twilight* .
0.894	*Heel* a golf ball .	*Heel* that dance .
0.887	*Create* a poem .	*Create* one a peer .
0.881	*Map* the genes .	*Map* the surface of Venus .
0.877	*Scallop* the hem of the dress .	*Scallop* the meat .
0.875	The senator received severe *criticism* from his opponent .	The politician received a lot of public *criticism* for his controversial stance on the issue .

Figure 4: False positives

[b]

cosine	sentence A	sentence B
0.092	*Hew* out a path in the rock .	One of the most widely used typefaces in the world was *hewn* by the English printer and typographer John Baskerville .
0.122	*Hail* a cab .	He was *hailed* as a hero .
0.193	*Bell* cows .	Who will *bell* the cat ?
0.225	*Boot* your computer .	When arriving at the office , first thing I do is *booting* my machine .
0.236	Let 's *peg* the rug to the floor .	*Peg* a tent .
0.237	*Gargle* with this liquid .	Every morning he *gargled* a little cheap Scotch .
0.258	Lydia put the change in her left *pocket* .	Lydia *pocketed* the change .
0.274	*Admit* someone to the profession .	She was *admitted* to the New Jersey Bar .
0.283	*Bag* a few pheasants .	We *bagged* three deer yesterday .
0.299	*Fold* up the newspaper .	Tony *folded* the flaps open .

Figure 5: False negatives

Figure 6: Top false positives and false negatives on baseline elmo2

[b]

cosine	sentence A	sentence B
0.986	He could not conceal his *hostility* .	He could no longer contain his *hostility* .
0.955	I could just make out her face in the *twilight* .	He loved the *twilight* .
0.95	*Engrave* a letter .	*Engrave* a pen .
0.946	*Indent* the documents .	*Indent* the paragraphs of a letter .
0.944	*Map* the genes .	*Map* the surface of Venus .
0.942	From the window he could catch a *glimpse* of the lake .	He caught only a *glimpse* of the professor 's meaning .
0.937	He lived in *exile* .	She lived as an *exile* .
0.927	A *growth* of hair .	The *growth* of culture .
0.927	The senator received severe *criticism* from his opponent .	The politician received a lot of public *criticism* for his controversial stance on the issue .
0.927	The *addition* of a leap day every four years .	The *addition* of a bathroom was a major improvement .

Figure 7: False positives

[b]

cosine	sentence A	sentence B
0.34	*Hew* out a path in the rock .	One of the most widely used typefaces in the world was *hewn* by the English printer and typographer John Baskerville .
0.343	*Hail* a cab .	He was *hailed* as a hero .
0.376	Please can I have a look , if I promise not to *touch* ?	Carrie *touched* his shoulder with the stick .
0.377	The neoclassical *canon* .	*Canons* of polite society .
0.377	The captain was obliged to *allowance* his crew .	Our provisions were *allowanced* .
0.413	Lydia put the change in her left *pocket* .	Lydia *pocketed* the change .
0.415	*Bell* cows .	Who will *bell* the cat ?
0.419	*Bag* a few pheasants .	We *bagged* three deer yesterday .
0.425	*Spend* money .	He *spends* far more on gambling than he does on living proper .
0.44	*Sign* an intersection .	This road has been *signed* .

Figure 8: False negatives

Figure 9: Top false positives and false negatives on VUA PoS

[b]

cosine	sentence A	sentence B
0.974	He could not conceal his *hostility* .	He could no longer contain his *hostility* .
0.888	*Scallop* the hem of the dress .	*Scallop* the meat .
0.878	He lived in *exile* .	She lived as an *exile* .
0.876	*Create* a poem .	*Create* one a peer .
0.868	*Indent* the documents .	*Indent* the paragraphs of a letter .
0.866	*Engrave* a letter .	*Engrave* a pen .
0.864	I could just make out her face in the *twilight* .	He loved the *twilight* .
0.859	An eyebrow *pencil* .	This artist 's favorite medium is *pencil* .
0.852	I *swear* by my grandmother 's recipes .	Before God I *swear* I am innocent .
0.842	The annual *crop* of students brings many new ideas .	The latest *crop* of fashions is about to hit the stores .

Figure 10: False positives

[b]

cosine	sentence A	sentence B
0.202	*Hew* out a path in the rock .	One of the most widely used typefaces in the world was *hewn* by the English printer and typographer John Baskerville .
0.276	The neoclassical *canon* .	*Canons* of polite society .
0.296	*Send* me your latest results .	Nora *sent* the book from Paris .
0.311	On entering a host cell , a virus will start to *replicate* .	*Replicate* the cell .
0.319	*Sign* an intersection .	This road has been *signed* .
0.345	*Hail* a cab .	He was *hailed* as a hero .
0.354	They had to *consult* before arriving at a decision .	*Consult* your local broker .
0.358	A generous *tipper* .	The Americans are among the most generous *tippers* in the world .
0.359	*Carve* one 's name into the bark .	That chisel *carved* the statue .
0.364	To *command* an army or a ship .	*Command* the military forces .

Figure 11: False negatives

Figure 12: Top false positives and false negatives on VUA SNLI