JOINT TEXT CLASSIFICATION ON MULTIPLE LEVELS WITH MULTIPLE LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Natural language uses words in an associative way to construct sentences: it is not words in isolation, but the appropriate use of hierarchical structures that makes communication successful. We propose a deep learning framework for explicitly tying together the representations between single words and full sentences, resulting in a fluid transfer of knowledge between these two levels of granularity. We construct a multi-head attention mechanism for sentence classification, where the individual attention heads simultaneously learn to perform multi-class sequence labeling. Supervision on individual tokens explicitly teaches the classifier which areas it needs to focus on in each sentence, while the sentence-level objective regularizes the token-level predictions and even enables sequence labeling without token-level training data. Our experiments show that the proposed architecture systematically outperforms its single-task counterparts and exhibits strong transfer capabilities, while also achieving reasonable performance as a zero-shot sequence labeler [1]

1 INTRODUCTION

Natural language has vast syntactic and semantic complexity: it involves using words in an associative way to construct structured sentences. Meaningful expressions are built from other meaningful expressions (Goldberg, 1995), or, as stated by the Principle of (Semantic) Compositionality (Hirst, 1987), the meaning of the whole is determined by the meaning of the parts. Neural architectures are often trained end-to-end, expecting the models to independently discover the necessary methods for language composition. However, due to data limitations on most tasks, composition patterns can be difficult to learn automatically. Furthermore, these discovered patterns can pick up noise and bias in the datasets and do not always match the desired model behavior.

We investigate a novel approach to applying the compositionality principle in a deep learning framework, allowing for more direct supervision. Our model learns to perform both sentence classification and token-level sequence labeling while tying these two objectives together in a way that enables them to reinforce each other. The network uses a multi-head attention mechanism to construct a sentence-level representation for multi-class sentence classification. At the same time, the customized attention heads are also used to construct token-level representations for multi-class sequence labeling. The token-level supervision explicitly teaches the classifier which areas it needs to focus on in each sentence, while the sentence-level objective regularizes the sequence labeling predictions.

Changpinyo et al. (2018) advocate for the beneficial integration of several, related tasks. In our case, the sentence and word predictions are explicitly tied together, so the network is incentivized to develop an organic relationship between hierarchies that resembles how humans use language. Moreover, as we formulate two distinct tasks based on the same dataset, we effortlessly provide the network with more training examples. We introduce different optimization objectives to satisfy the aim of each task – this implicitly regularizes the weights towards better (i.e., more general) text representations (Ruder, 2017). Despite these theoretical guarantees, we seek to empirically determine whether (and how) the interaction between the two levels of granularity benefits learning.

[1] Code available at `https://github-placeholder`.

Recent work has shown promising results for directly supervising the internal components of a model. For example, Liu et al. (2016) used alignment annotations to improve the performance of a neural machine translation system. Rei & Søgaard (2019) described an architecture for supervising attention in a binary text classification setting. Barrett et al. (2018) used a related model to guide the network to focus on similar areas as humans, based on human gaze recordings. We build on these ideas and describe a more general framework, extending it to both multi-class text classification and multi-class sequence labeling.

We propose the following training conditions and experimental settings, grouped by the amount of word-level annotation used by our model to guide its learning:

- **Fully supervised**: the system is provided with full annotations both on the sentence and on the token level. The model has all the information needed to perform very well on each isolated task. However, we are more interested in how the multi- and single-task performances compare. Does the model take advantage of the joint learning regime and the supplemental labeled data to increase its performance on each task?
- **Semi-supervised**: the system is provided with some supervision signal, but only for a subset of the tokens, while sentences are always receiving it in full. We investigate our model's inference abilities and determine the proportion of token annotation that is sufficient for the network to reach as good a performance as the fully supervised one.
- **Unsupervised**: learning sequence labeling without any token-level annotations (*zero-shot sequence labeling*). In other words, we train a sentence classifier and evaluate it as a sequence labeler. If knowledge can be transferred from a higher, abstract sentence-level to a lower, fine-grained token-level, the system will perform sophisticated word-predictions solely based on the considerably cheaper sentence annotations.

We augment our model with several auxiliary objectives and add a new regularization term to incentivize the construction of distinct tag-specific sub-spaces. As we intertwine the two hierarchical levels, our network exhibits strong transfer capabilities that we validate on three different tasks.

2 THE MULTI-HEAD ATTENTION LABELER (MHAL)

Our architecture is designed to tie the token and sentence representations together. It has two core components: the first one uses Bi-LSTMs to build compact vectors for each word; the second uses a multi-head attention mechanism to obtain two distributions, over the tagset and the sentence labels. The goal of the mechanism is to perform sentence classification. However, because each head acts as a separate label predictor, the system also behaves like a sequence labeler: the scores based on which we make sentence-level predictions are obtained by combining the individual attention weights, which are attached to each word and used to make token-level predictions. Joining multiple levels in this way to detect multiple labels is one of our main contributions.

The first component takes as input a tokenized sentence of length N and maps it to a sequence of vectors $[x_1, x_2, ..., x_N]$. Each vector x_i, corresponding to the i^{th} token in a sentence, is the concatenation of its pre-trained word embedding w_i with its character-level representation c_i, similar to Lample et al. (2016). Passing each vector x_i to a Bi-LSTM (Graves & Schmidhuber, 2005), we obtain compact token representations z_i by merging the hidden states from each direction at every time step and projecting these onto a joint feature space using a *tanh* activation (equations 1 to 3).

$$\overrightarrow{z_i} = \text{LSTM}(x_i, \overrightarrow{z_{i-1}}) \tag{1}$$

$$\overleftarrow{z_i} = \text{LSTM}(x_i, \overleftarrow{z_{i+1}}) \tag{2}$$

$$z_i = tanh(W_z[\overrightarrow{z_i}; \overleftarrow{z_i}] + b_z) \tag{3}$$

The second component of our architecture is a multi-head attention mechanism (Vaswani et al., 2017) that creates H heads. We set H equal to the size of our tagset and create a correspondence between attention heads and token labels. For each continuous vector representation z_i, and for each head $h \in \{1, 2, ..., H\}$, we obtain three vectors of keys, queries and values (denoted by k_{ih}, q_{ih}, and v_{ih}, respectively) by non-linearly projecting z_i onto a different sub-space that is H times smaller

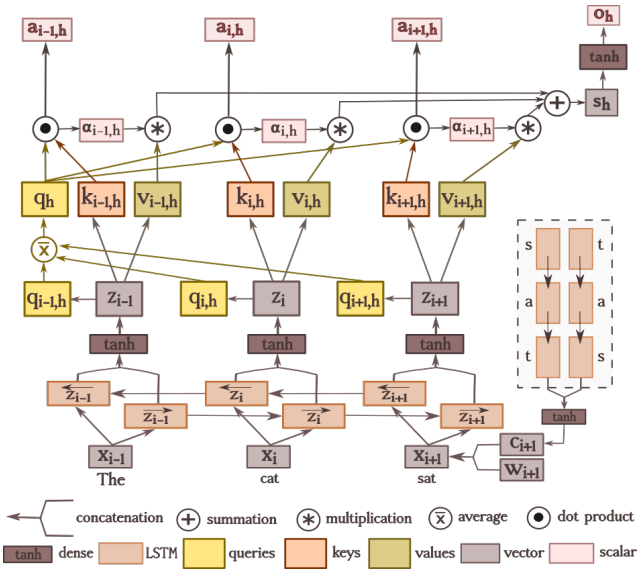

Figure 1: MHAL architecture, taking a sequence of three words. Illustrated for one head h only. We obtain character representations as presented in the dashed rectangle (here, for the word *sat*).

than the original input size. A critical difference between our approach and the standard multi-head attention mechanism is that we collapse the queries q_{ih} (operating over each token i and each head h) into a single query q_h that is specific to head h. We achieve this by averaging q_{ih} across the sentence, as in equation 4. The motivation is to obtain a compressed, individual representation that is shared across the sentence but still encapsulates the identity of a single, specific tag.

We define our attention function as the dot product (denoted by $\bullet$) between a query and its corresponding key, resulting in the attention evidence scores $a_{ih} \in \mathbb{R}^1$. We apply a *sigmoid* (σ) activation function[2] and normalize the attention scores across the words to obtain the attention weights $\alpha_{ih} \in [0, 1]$. Next, similar to Yang et al. (2016), we determine the importance of each value vector by multiplying it with its corresponding attention weight. Summing these over the words in a sentence, we get a representation $s_h \in \mathbb{R}^d$, which is further passed through two feedforward layers: the first one is non-linear and projects the sentence representation onto a smaller feature space, while the last one is linear and outputs a scalar sentence score o_h for each head h (equations 5 to 8).

$$q_h = \frac{1}{N}\sum_{i=1}^{N} q_{ih} \quad (4) \qquad a_{ih} = q_h \bullet k_{ih} \quad (5)$$

$$\alpha_{ih} = \frac{\sigma(a_{ih})}{\sum_{j=1}^{N}\sigma(a_{jh})} \quad (6) \qquad s_h = \sum_{i=1}^{N} \alpha_{ih} v_{ih} \quad (7)$$

$$o_h = W_o tanh(W_s s_h + b_s) + b_o \quad (8)$$

where W_o and W_s are weight matrices, and b_o and b_s are bias vectors.

We need to collect the sentence scores across all heads and make a sentence prediction. The challenge arises as we have to map these H scores (equal to the number of token labels) to the number of sentence labels S, which are not necessarily in direct correspondence. To solve this, we use the fact that datasets typically have a *default label* that is common between the token and the sentence label sets and that vastly outnumbers the others. We distinguish two situations:

[2]We chose the *sigmoid* over the *softmax* to impose a smoother distribution over the tokens attended.

1. $\boldsymbol{H = S}$: Each sentence label has a corresponding word-tag (and thus, a head associated). We can concatenate the sentence scores across all heads into a vector $\tilde{o} = [o_1; o_2; ...; o_H]$.
2. $\boldsymbol{H \neq S}$ and $\boldsymbol{S = 2}$: The sentence labels are binary, while the token labels are not, so we have to find a correspondence between the heads and the two sentence labels. We concatenate the score obtained for the default head o_d (corresponding to the default label) with the maximum score obtained for the non-default heads o_{nd}: $\tilde{o} = [o_d; o_{nd}]$, where d and nd are the indices of the default and non-default heads, respectively, and $o_{nd} = \max_{h \neq d}(o_h)$.

We obtain a normalized distribution $\tilde{y}$ over the sentence labels by applying a *softmax* on the extracted scores $\tilde{o} \in \mathbb{R}^S$ and predict the label corresponding to the most probable sentence score.

In addition to sentence classification, we also want to make token-level predictions. To achieve this, we treat each attention evidence a_{ih} as the score of an individual word i and head h. It is crucial to note that the sentence scores (and thus, predictions) rely on the attention evidence – we conditioned the sentence label distribution $\tilde{y}$ on the token scores. Thus, the sentence and token-level predictions are intertwined, and we can perform sequence labeling in tandem with sentence classification by re-using the attention evidence scores and re-interpreting them as token-specific predictions. By explicitly tying together these two levels of granularity, we incentivized the network to learn better composition functions and share language features between layers. Finally, we apply a *softmax* on the concatenation of scores across all heads to obtain the token label distribution $\tilde{t_i} \in \mathbb{R}^H$ and predict the tag with the maximum probability.

This subsumes the architectural design of our joint text classifier, to which we refer to as the multi-head attention labeler (MHAL), schematically represented in Figure 1.

3 Optimization objectives

Our model can be optimized both as a sentence classifier and as a sequence labeler. Both losses, L_{sent} and L_{tok}, minimize the summation over the categorical cross-entropy between the predicted sentence (or token) label distribution $\tilde{y}$ (or $\tilde{t_i}$) and its true annotation y (or its true gold tag t_i):

$$L_{sent} = -\sum_{s}\sum_{j=1}^{S} y_j^{(s)} log(\tilde{y}_j^{(s)}) \tag{9}$$

$$L_{tok} = -\sum_{s}\sum_{i=1}^{N}\sum_{j=1}^{H} t_{ij}^{(s)} log(\tilde{t}_{ij}^{(s)}) \tag{10}$$

where $y_j^{(s)}$ and $t_{ij}^{(s)}$ are binary indicator variables specifying whether sentence s truly is a sentence of label j and token t at position i in sentence s truly is a token of tag type j, respectively.

Recall that the sentence label distribution is based on the attention evidence scores, which represent, in turn, the token scores used for word-level classifications. If we train our model solely as a sentence classifier (by providing only sentence-level annotations), the network will also optimize the token scores because the parameter updates on the sequence labeling task are performed at layers below the sentence classification task. Moreover, the network will learn the important areas of a sentence, combining the scores from individual words to determine the overall sentence label. In this way, our model performs zero-shot sequence labeling, a type of transductive transfer learning (Ruder, 2017). In contrast, optimizing only the parameters used in the sequence labeling task does not implicitly train the sentence scores since they are situated above it in the architecture. However, when both levels receive supervision, the token signal encourages the network to put more weight on the attention heads indicative of the correct labels.

Previous research in multi-task learning provides significant evidence that including several related tasks along with the core task positively impacts performance due to the regularization and generalization effects of the procedure (Bingel & Søgaard, 2017; Søgaard & Goldberg, 2016; Changpinyo et al., 2018). Closely following the setting proposed by Rei (2017), we include language modeling (LM) as a secondary objective, operating both over characters and words. In this way, we inject

corpus-specific information into the model as well as syntactic and semantic patterns (Linzen et al., 2016; Marvin & Linzen, 2018) and expect them to lead to more transferable features.

We consider another auxiliary loss, whose purpose is to better wire the two granularity levels, of sentences and words. We call it *attention objective* because it directly operates on the attention heads corresponding to the correct sentence label, imposing two conditions:

1. There should be at least one word of the same label as the ground-truth sentence. Intuitively, most of the focus should be on the words indicative of the sentence type.
2. There should be at least one word that has a default label. Even if the sentence has a non-default class, it should still contain at least one default word.

These conditions can be formulated as a loss function and then optimized during training:

$$L_{attn} = \sum_s \big(\max_i(\tilde{t^*}^{(s)}_{i,h=k}) - q'_k \big)^2 + \sum_s \big(\max_i(\tilde{t^*}^{(s)}_{i,h=d}) - q'_d \big)^2 \tag{11}$$

where

$$\tilde{t^*}_{i,h} = \begin{cases} \tilde{t}_{i,h}, & \text{if } (H = S) \vee (h = d \wedge H \neq S \wedge S = 2) \\ \max\limits_{j \neq d}(\tilde{t}_{i,j}), & \text{if } (h \neq d \wedge H \neq S \wedge S = 2) \end{cases} \tag{12}$$

and d is the default head, k is the true sentence label, $\tilde{t}_i$ is the predicted token label distribution for word i (so $\tilde{t}_{i,h}$ is its value for head h), and, finally, q' is the smoothed sentence label distribution (so q'_j is its value for sentence label j), whose expression is obtained as in Szegedy et al. (2016), choosing $\epsilon = 0.15$ and $K = H$.

Lastly, we propose a custom regularization term for the multi-head attention mechanism to motivate the network to learn a truly distinct representation sub-space for each of the query vectors q_h. As opposed to the keys and values, which are associated with (possibly reoccurring) words, the queries q_h encapsulate the essence of a certain tag. Thus, they need to capture the distinctive features that are specific to a particular head. To push the network towards this goal, we introduce the term R_q and calculate it as the average cosine similarity between every pair of queries q_h and q_i, with $h \neq i$ (equation 13). R_q can be viewed as an orthogonality penalty: it attains a minimum for two query vectors spanning over orthogonal sub-spaces, and a maximum when their sub-spaces coincide. Thus, this technique imposes a wider angle between the queries, encouraging the model to learn unique, diverse, and meaningful vector representations.

$$R_q = \frac{2}{H(H-1)} \sum_{h=1}^{H-1} \sum_{i>h}^{H} \frac{q_h \cdot q_i}{\|q_h\| \cdot \|q_i\|} \tag{13}$$

The final loss function L_{tot} is a weighted sum of our objectives, allowing us to observe the effect of the different components as well as to control the flow of the supervision signal and the importance of each auxiliary task: $L_{tot} = \lambda_{sent} L_{sent} + \lambda_{tok} L_{tok} + \lambda_{LM} L_{LM} + \lambda_{attn} L_{attn} + \lambda_{R_q} R_q$.

4 EXPERIMENTS

Our model was evaluated on three different datasets: Stanford Sentiment Treebank (SST, Socher et al., 2013) for sentiment analysis, CoNLL-2003 (Tjong Kim Sang & De Meulder, 2003) for named entity recognition (NER), and the First Certificate in English (FCE, Yannakoudakis et al., 2011) for fine-grained grammatical error detection. All of them have already been tokenized and split into train, development, and test sets. In the appendix, we provide some corpus statistics (Table 4) and a detailed breakdown of the number of annotated examples available per split and per label (Table 5).

Our datasets contain sentences with labeled words, used to make token-level predictions. Thus, they can be phrased as sequence labeling tasks: for SST, identifying the positive, negative and

neutral words; for CoNLL-2003, detecting the persons, organizations, locations, miscellaneous, and the non-named entities; for FCE, identifying the errors in content, form, function, orthography, or others as well as the correct words. Given a sequence of labeled words, there is an implicit label for the sentence too. For instance, a sentence containing at least one grammatical mistake makes it ungrammatical overall. SST provides annotations for each sentence in the corpus, using the same labels as for the tagset. For CoNLL-2003 and FCE, there are no pre-existing sentence-level annotations, but we can infer a binary label for each sentence based on the existence of at least one word annotated as an entity or as a grammatical error, respectively. As already mentioned, each dataset has a default label, common between the token and the sentence label sets, and, in our case, it corresponds to neutral phrases, non-named entities, and grammatical words, respectively.

We did not engage in fine-tuning our neural network's components and largely followed the settings proposed by Rei & Søgaard (2019). We manually searched the best values for our new parameters based on the performance on the development set (see Table 6 in the appendix for a complete list). To avoid outliers, we ran each experiment with five different random seeds and reported the mean results using the metrics specific to each dataset along with the mean micro-average score (denoted by a subscript μ) of the non-default labels (denoted by a superscript $*$), as commonly used in the multi-task learning literature (Changpinyo et al., 2018; Martínez Alonso & Plank, 2017).

We train our models under the different regimes mentioned in the introduction by assigning corresponding values to the weights λ in the expression of L_{tot}. We distinguish the following models:

- **MHAL-sent**: single-task sentence classifier; $\lambda_{sent} = 1.0$, while all the other λs are zero.
- **MHAL-tok**: single-task sequence labeler; $\lambda_{tok} = 1.0$, while all the other λs are zero.
- **MHAL-sent+tok**: optimized both as a sentence classifier and a sequence labeler; $\lambda_{sent} = \lambda_{tok} = 1.0$, while all the other λs are set to zero.
- **MHAL-joint**: just like MHAL-sent+tok, it performs symmetric multi-task learning. In addition, it also sets $\lambda_{attn} = 0.01$, $\lambda_{LM} = 0.1$, and $R_q = 0.5$.
- **MHAL-zero**: zero-shot sequence labeling where only sentence-level annotation is available for training; setting $\lambda_{sent} = 1.0$, $\lambda_{tok} = 0.0$, and all the other λs as in MHAL-joint.

	SST				**CoNLL-2003**				**FCE**			
	P^*_μ	R^*_μ	$F^*_{1\mu}$	Acc	P^*_μ	R^*_μ	$F^*_{1\mu}$	$F_{1\mu}$	P^*_μ	R^*_μ	$F^*_{1\mu}$	$F_{1\mu}$
Random	38.99	30.81	34.42	31.58	79.11	48.91	60.45	48.91	67.64	51.16	58.26	51.07
MHAL-sent	71.40	83.75	77.08	70.12	97.16	**99.20**	98.17	97.06	79.46	89.09	84.00	77.58
MHAL-zero	71.61	83.80	77.23	**71.08**	97.18	98.80	97.98	96.80	77.96	**93.41**	84.99	77.90
MHAL-sent+tok	**72.23**	83.14	77.30	70.14	**97.92**	99.09	**98.50**	**97.53**	**85.61**	84.66	85.13	78.92
MHAL-joint	71.34	**84.90**	**77.53**	70.24	97.82	99.13	98.47	97.32	83.58	86.82	**85.17**	**79.50**

Table 1: Sentence classification performance on SST, CoNLL-2003, and FCE datasets.

In Table 1, we present the performance of our models when trained as sentence classifiers. For SST and FCE, MHAL-zero outperforms MHAL-sent, suggesting that MHAL-zero's additional auxiliary objectives further refine the predictions. Comparing these single-task sentence classifiers to their multi-task counterparts (MHAL-sent+tok and MHAL-joint, which have also received token-annotated examples), we observe that the $F^*_{1\mu}$ score consistently increases for the latter. This finding suggests that the additional information extracted from individually labeled words was successfully transferred to the sentence-level task. Therefore, MHAL-sent+tok and MHAL-joint become exponents of the usefulness of multi-task learning, being our best models.

In Table 2, we present the performance of our models when trained as sequence labelers. Our zero-shot sequence labeling model MHAL-zero exploits the generality of neural networks, sharing features from a higher to a lower level task, and performs the best amongst other models that do not use token-level supervision. For instance, the random label assignment can be a strong baseline, particularly for highly skewed datasets, such as FCE; MHAL-zero surpasses it, while the simple sentence classifier MHAL-sent does not always do so. MHAL-zero persistently outperforms MHAL-sent by a large margin on all datasets (e.g. it gave a 16.31% boost in $F^*_{1\mu}$ on SST). Since the difference between them resides in the activated objectives, these results show that our auxiliary losses introduce a necessary inductive bias, improving MHAL performance.

	SST				CoNLL-2003				FCE			
	P^*_μ	R^*_μ	$F^*_{1\mu}$	$F_{1\mu}$	C-P	C-R	C-F_1	$F^*_{1\mu}$	$P^*_{0.5\mu}$	$R^*_{0.5\mu}$	$F^*_{0.5\mu}$	$F_{0.5\mu}$
Random	11.01	33.59	16.59	26.41	13.09	20.14	15.87	7.20	2.60	16.78	4.50	16.63
MHAL-tok	87.20	70.90	78.19	92.13	90.75	91.47	91.11	90.31	44.48	15.89	23.34	87.36
MHAL-sent	8.67	13.85	10.62	56.35	13.18	21.95	16.47	9.52	2.46	15.85	4.24	35.33
MHAL-zero	21.60	39.78	26.93	67.26	24.02	27.23	25.51	21.54	4.03	12.00	5.64	60.84
MHAL-sent+tok	**87.52**	72.35	79.21	92.21	**91.05**	91.69	91.37	**90.77**	43.41	17.45	24.47	87.97
MHAL-joint	87.47	**73.12**	**79.65**	**92.37**	91.02	**91.75**	**91.38**	90.70	**45.66**	**20.65**	**28.25**	**87.98**

Table 2: Sequence labeling performance on SST, CoNLL-2003, and FCE datasets. Metrics starting with *C-* represent span-sensitive scores, specific to the NER task.

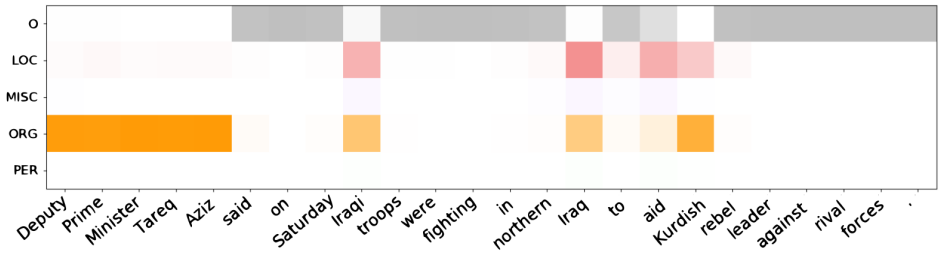

Figure 2: Attention evidence scores, normalized acrossed heads, assigned by MHAL-zero for the words in a sentence from the CoNLL-2003 dataset.

Visualization of the MHAL-zero attention heads can provide a way to understand the patterns learned by the network. Figure 2 shows a sentence from the CoNLL-2003 dataset along with its attention evidence scores, normalized across the heads. The model is able to identify entities in the sentence even when trained with only a binary sentence-level signal. However, this is a difficult task and it does not always assign the correct entity type, here mistaking the person for an organization. By including token-level supervision into the model, MHAL-joint is able to further improve on this result. We include additional visualizations of both MHAL-zero and MHAL-joint in the appendix.

Note that our multi-task models (MHAL-sent+tok, MHAL-joint) register systematic improvements across all our datasets over the single-task sequence labelers, which were trained in isolation (MHAL-tok, MHAL-sent), further emphasizing the effectiveness of sharing information between the two granularity levels. On our way to obtaining the sentence scores, we re-interpreted each attention evidence as a token score. Now, based on the results presented, it follows that wiring each head to a classifier designated for a single label has better stimulated the network to learn shared representations and excel at performing two tasks, under the same architecture.

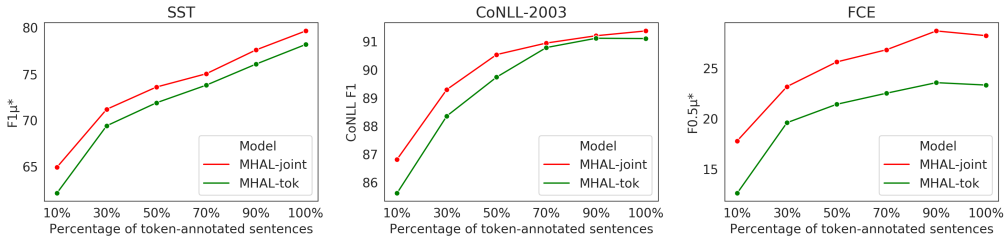

Figure 3: Semi-supervised experiments for SST, CoNLL-2003, and FCE, comparing the sequence labeling performance of the multi-task model MHAL-joint with the single-task model MHAL-tok.

We perform a semi-supervised experiment on MHAL-joint, using the supervision signal of all sentences and only a percentage p of the word-level annotations. In Figure 3, we present the sequence labeling results of our multi-task MHAL-joint, in comparison to the single-task MHAL-tok, gradually increasing p to allow more tokens to guide learning. We observe that adding as little as 10% of the token-annotated sentences increases MHAL-zero's performance by substantial amounts (38.01% on SST, 61.31% on CoNLL-2003, and 12.15% on FCE), suggesting that the two tasks are positively

influencing each other. Using only 30% of the data, MHAL already approaches its fully-supervised performance, showing that the transfer of knowledge has benefits that flow in both directions between sentences and words. Compared to MHAL-tok, MHAL-joint is systematically better – the multi-task model has the annotated sentences at its disposal and learns to use them to its advantage particularly when receiving weak token supervision signals (steep increases for $\leq 50\%$).

Dev metric	**SST**				**CoNLL-2003**				**FCE**			
	P^*_μ	R^*_μ	$F^*_{1\mu}$	S-Acc	C-P	C-R	C-F_1	S-F_1	P^*_μ	R^*_μ	$F^*_{0.5\mu}$	S-F_1
S-$F^*_{1\mu}$	21.60	39.78	26.93	**71.08**	24.02	27.23	25.51	96.80	**4.03**	12.00	5.64	77.90
$F^*_{1\mu}$	23.21	32.97	27.24	68.64	20.03	24.25	21.79	93.05	3.56	**18.52**	5.96	75.83
(S-$F^*_{1\mu}$+$F^*_{1\mu}$)/2	**23.34**	**47.00**	**30.22**	70.92	**24.10**	**28.02**	**25.92**	**96.90**	3.85	17.12	**6.28**	**78.03**

Table 3: The effect of the stopping criterion metric during the training of MHAL-zero.

Across all our experiments, we impose two stopping criteria and apply them on the development set: **1.** the sentence-level classification performance (S-$F^*_{1\mu}$), adopted by all models that do not receive any token-level annotation (e.g. MHAL-zero); **2.** the token-level classification performance ($F^*_{1\mu}$), adopted by all models that receive some token annotation (e.g. MHAL-tok, MHAL-joint).

We observed that, even in the case of MHAL-zero, stopping based on the token performance improves the word-level predictions at test time, but usually hurts the sentence predictions. However, as suggested by the results in Table 3, stopping based on the average of these two metrics generally improves both the token and the sentence predictions.

The network usually takes more time to reach the common optimal point when we include the token-based stopping criterion. Sentence classification is an easier task than sequence labeling – being predicted at a higher layer in the network hierarchy, it accumulates more information and thus builds solid abstractions, not to mention that it has fewer unique labels. For these reasons, the network falls into a local minimum when guided by the sentence-level performance. However, choosing tokens as a stopping criterion requires annotated development data, which does not generally comply with the framing of our zero-shot learning experiment. Nevertheless, reporting this finding emphasizes that the stopping criterion requires careful consideration – it is responsible for choosing the best performing model used during testing and for driving the application of the learning rate decay. A few performance percentage points could be gained by carefully selecting the stopping metric.

5 Conclusion

In this paper, we proposed MHAL, a novel model that ties together two hierarchical levels (for sentences and words) to build representations whose richness and utility were evaluated on three different tasks: sentiment analysis, named entity recognition, and grammatical error detection. MHAL's architecture is based on a multi-head attention mechanism that re-interprets each head as an individual label classifier, treating each attention evidence as a token score, based on which we make multi-class word-level predictions. The attention weights also contribute to making sentence predictions, so the two granularity levels are intertwined and incentivized to help each other by sharing knowledge. Therefore, through this design innovation, our model can perform two prediction problems – sequence labeling and sentence classification – using the same, shared architecture.

We proposed training MHAL under various regimes; the results confirm its versatility and robustness. The several auxiliary objectives introduced (attention loss, language modeling, query regularization), whose purpose was to be "bridges" between the two prediction tasks, proved to be beneficial additions. Our zero-shot sequence labeling model learned suggestive patterns for the whole sentence and successfully transferred them to the lower-level task. Helped by their architectural design, our multi-task models (MHAL-sent+tok and MHAL-joint) always outperform their single-task counterparts (MHAL-sent and MHAL-tok). This finding is important as it reveals the benefits of cooperation between related tasks: conditioning the sentence scores on the unnormalized attention weights, which are, in turn, used as token scores, caused a fluid flow of information and the development of an organic relation between granularity levels. Therefore, by (re-)designing joint models, we can bring NLP transfer learning closer to performing structured multi-level text understanding and labeling.

REFERENCES

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 302–312. Association for Computational Linguistics, 2018.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 164–169. Association for Computational Linguistics, 2017.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2965–2977. Association for Computational Linguistics, 2018.

Adele Goldberg. Constructions: A construction grammar approach to argument structure. *The University of Chicago Press*, 1995.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pp. 5–6, 2005.

Graeme Hirst. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, New York, NY, USA, 1987.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawa-kami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the Conference of the NAACL-HLT*, pp. 260–270. Association for Computational Linguistics, 2016.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3093–3102, Osaka, Japan, 2016. The COLING 2016 Organizing Committee.

Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 44–53. Association for Computational Linguistics, 2017.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202. Association for Computational Linguistics, 2018.

Marek Rei. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 2121–2130. Association for Computational Linguistics, 2017.

Marek Rei and Anders Søgaard. Jointly learning to label sentences and tokens. In *Proceedings of the 33rd National Conference on Artifical Intelligence*, pp. 6916–6923. AAAI Press, 2019.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013.

Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 231–235. Association for Computational Linguistics, 2016.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147. Association for Computational Linguistics, 2003.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489. Association for Computational Linguistics, 2016.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 180–189. Association for Computational Linguistics, 2011.

A APPENDIX

Dataset	**No. labels**		**Prop. O**		**Full Entropy**		**Non-O Entropy**	
	sent	tok	sent	tok	sent	tok	sent	tok
SST	3	3	0.19	0.78	1.509	0.961	0.999	0.956
CoNLL-2003	2	5	0.20	0.83	0.731	0.979	0.263	1.929
FCE	2	6	0.37	0.89	0.952	0.775	0.421	2.288

Table 4: We list, for sentences and tokens: number of unique labels, proportion of default labels (O), entropy of the label distribution, and entropy of the non-default label distribution (using $\log_2$).

Dataset	**Label**	**Number of sentences**			**Number of tokens**		
		Train	Dev	Test	Train	Dev	Test
SST	O	1,624	229	389	128,156	16,684	33,128
	N	3,310	428	912	13,384	1,740	3,488
	P	3,610	444	909	22,026	2,850	5,789
	Total	8,544	1,101	2,210	163,566	21,274	42,405
CoNLL-2003	O	2,909	645	697	169,578	42,759	38,323
	LOC	11,132	2,605	2,756	8,297	2,094	1,925
	MISC				4,593	1,268	918
	ORG				10,025	2,092	2,496
	PER				11,128	3,149	2,773
	Total	14,041	3,250	3,453	203,621	51,362	46,435
FCE	O	10,718	824	900	396,479	30,188	35,525
	CONTENT	17,836	1,384	1,806	7,194	527	673
	FORM				8,174	621	850
	FUNC				11,084	888	1,194
	ORTH				12,655	1,126	1,429
	OTHER				11,415	861	1,116
	Total	28,554	2,208	2,706	447,001	34,211	40,787

Table 5: Statistics of the labeled sentences and tokens, separated by the train, dev or test split.

Hyperparameter	Value	Description
word embedding size	300	Size of the word embeddings.
char embedding size	100	Size of the character embeddings.
word recurrent size	300	Size of the word-level Bi-LSTM hidden layers.
char recurrent size	100	Size of the character-level Bi-LSTM hidden layers.
word hidden layer size	50	Compact word vector size, applied after the last Bi-LSTM.
char hidden layer size	50	Char representation size, applied before concatenation.
attention evidence size	100	Layer size for predicting attention weights.
hidden layer size	200	Final hidden layer size, right before word-level predictions.
max batch size	32	Number of sentences taken for training.
epochs	200	Maximum number of epochs to run the experiment for.
stop if no improvement	7	Stop if there has been no improvement for this many epochs.
learning rate	1.0	The learning rate used in AdaDelta.
decay	0.9	Learning rate decay used in AdaDelta.
input dropout	0.5	Value of the dropout applied after the LSTMs.
attention dropout	0.5	Value of the dropout applied on the attention mechanism.
LM max vocab size	7500	Max vocabulary size for the language modeling objective.
smoothing epsilon	0.15	The value of the epsilon in label smoothing.
stopping criterion	$F^*_{1\mu}$	The development metric used as the stopping criterion.
optimization algorithm	AdaDelta	Optimization algorithm used.
initializer	Glorot	Method for random initialization.

Table 6: Hyperparameter settings for all of our MHAL models.

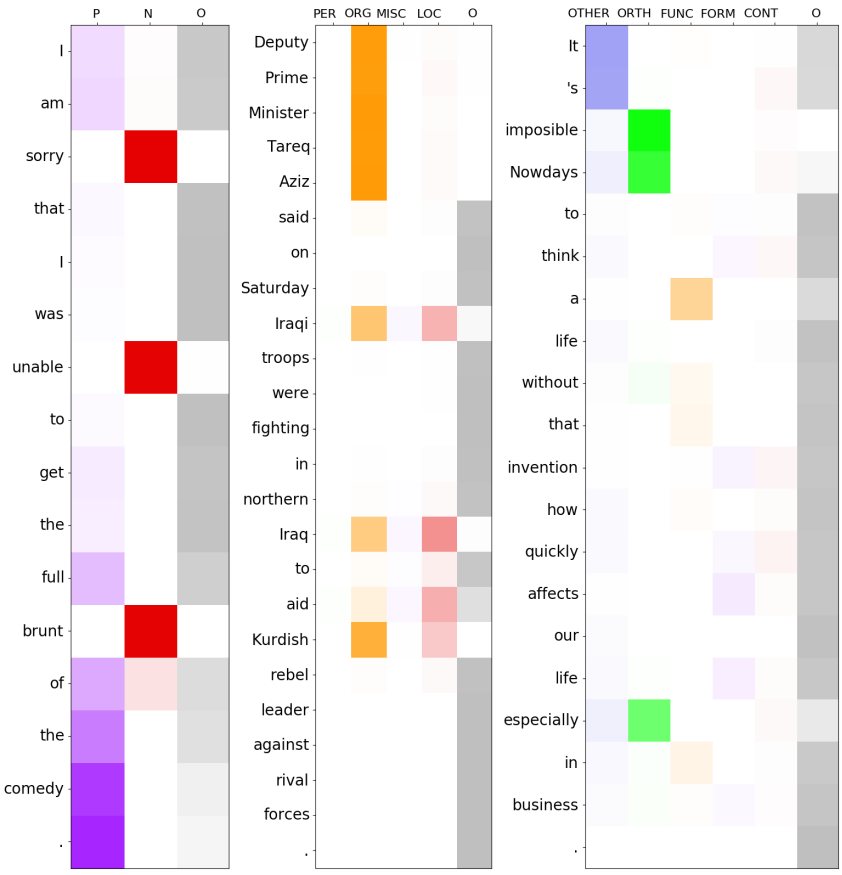

Figure 4: Attention evidence scores, normalized acrossed heads, assigned by MHAL-zero for the words in three sentences from the SST (leftmost), CoNLL-2003 (middle), and FCE (rightmost) datasets.

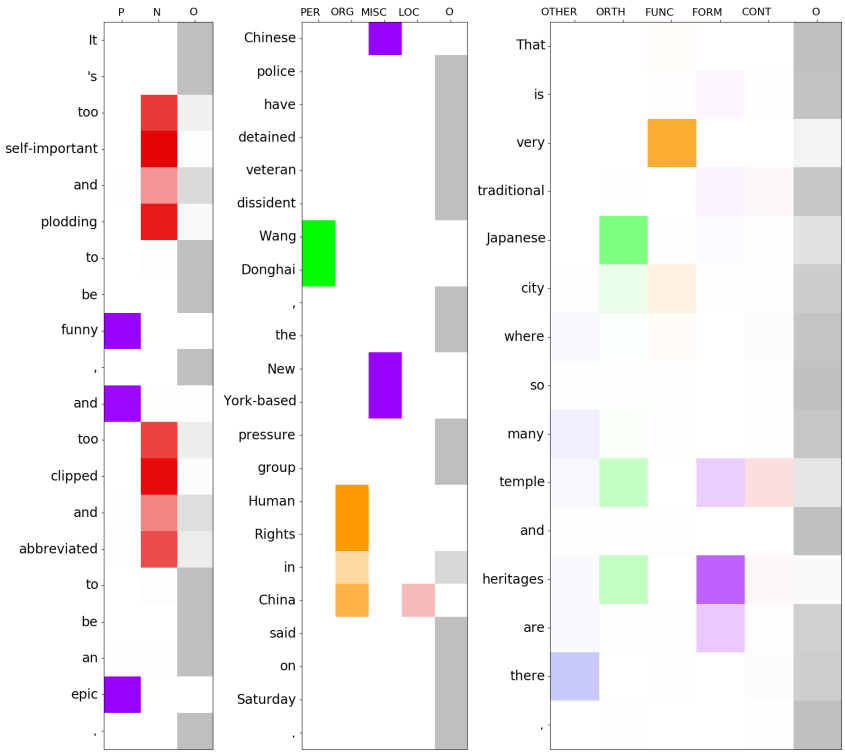

Figure 5: Attention evidence scores, normalized acrossed heads, assigned by MHAL-joint for the words in three sentences from the SST (leftmost), CoNLL-2003 (middle), and FCE (rightmost) datasets.