Understanding the Representational Power of Neural Retrieval Models Using NLP Tasks

Daniel Cohen, Brendan O'Connor, W. Bruce Croft
University of Massachusetts Amherst
Amherst, MA
{cohen,brenocon,croft}@cs.umass.edu

ABSTRACT

The ease of constructing effective neural networks has resulted in a large number of varying architectures iteratively improving performance on a task. Due to the nature of these models being black boxes, standard weight inspection is difficult. We propose a probe based methodology to evaluate what information is important or extraneous at each level of a network. We input natural language processing datasets into a trained answer passage neural network. Each layer of the neural network is used as input into a unique classifier, or probe, to correctly label that input with respect to a natural language processing task, probing the internal representations for information. Using this approach, we analyze the information relevant to retrieving answer passages from the perspective of information needed for part of speech tagging, named entity retrieval, sentiment classification, and textual entailment. We show a significant information need difference between two seemingly similar question answering collections, and demonstrate that passage retrieval and textual entailment share a common information space, while POS and NER information is used only at a compositional level in the lower layers of an information retrieval model. Lastly, we demonstrate that incorporating this information into a multitask environment is correlated to the information retained by these models during the probe inspection phase.

ACM Reference Format:
Daniel Cohen, Brendan O'Connor, W. Bruce Croft. 2018. Understanding the Representational Power of Neural Retrieval Models Using NLP Tasks. In *2018 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '18), September 14–17, 2018, Tianjin, China.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3234944.3234959

1 INTRODUCTION

An effective information retrieval (IR) system must be able to process a query and candidate document and produce an accurate relevance score, which necessitates the model be able to capture the complex relations between its query and candidate document inputs. As conventional IR features do not adequately capture this signal in certain retrieval domains such as short text and passage retrieval [6, 26], neural networks provide a strong foundation as retrieval models due to their capability to learn features via backpropagation. In particular, recurrent neural networks have been shown to achieve state of the art performance on these domains [6, 32].

As handcrafted features perform particularly poorly for these IR tasks, the relevant features therefore rely on the neural model's multiple nonlinear transformations across spans of text to reach a salient representation. This black box nature results in the publication of numerous models with only the final rankings used for benchmarking, and it is further exacerbated by the impact of hyperparameters on the performance of a single network [9] as well as our lack of understanding in how these models generalize [35].

Recent work has sought to alleviate this opaque nature by leveraging attention mechanisms [17], gradients [14], or cell activations [11] to understand what information is important. However, these methods rely on the original input to understand what is occurring rather than viewing the information in context of a structured task and only identify topical representations or patterns in text that correspond with a strong prediction for a label. When applied to an IR task, these techniques reveal traditional handcrafted features such as term overlap and related words, but provide little information on what non traditional features are being captured to produce a relevance score [22, 23].

Thus we propose viewing IR neural models as transformations on traditional NLP information defined through auxiliary tasks rather than at the traditional term level. This approach portrays relevance features as a function of distributed NLP features, and while this is not innately interpretable, the rigid definition of these auxiliary tasks provide a reference to examine how the IR model leverages these non traditional features. To demonstrate the degree to which relevance is composed of NLP information, we provide a new technique leveraging Alain and Bengio's [2] probe based methodology and apply it to measure the information loss with respect to related tasks. As NLP objectives are often highly structured, they provide a more concrete environment to understand what information is being captured in a network trained for retrieval. To do this, we insert small neural networks, referred to as probes, into the intermediate layers of a neural model trained for IR and evaluate the hidden representations for auxiliary NLP applications. As the final output of an IR model is a single scalar, this probe based methodology allows one to see what NLP features are most important for lower level layers as well as how the IR model learns to combine or discard this information into new representations pertinent to determining relevance. The motivation for this approach is three fold: (1) Understanding the NLP information pertinent to the core retrieval task allows for IR researchers to leverage the abundant work done with respect to that specific NLP application when designing new models, (2) Providing insight into what low level and high level features an IR network

ICTIR '18, September 14–17, 2018, Tianjin, China

ACM ISBN 978-1-4503-5656-5/18/09…$15.00
https://doi.org/10.1145/3234944.3234959

leverages to produce a relevance score in order to better engineer future neural architectures, and (3) Identifying common patterns on what information is needed to determine relevance across IR collections.

We choose answer passage retrieval as a case study of this technique, although it is generalizable to any retrieval task. The nature of answer passage retrieval is conducive to the difficulty that traditional retrieval techniques experience and the recent success of deep neural networks on this task [7].

Through this technique, we show that increasingly abstract features are learned at subsequent layers within an IR network, and correspondingly that lower level NLP tasks such as POS tagging and NER suffer the greatest at higher hidden layers when compared to identifying textual entailment, a higher level NLP task. We evaluate this over two similar retrieval collections and reveal a significant difference in information used within their respectively trained neural models. This confirms that sentence retrieval is a significantly different task than passage retrieval in terms of useful neural representations. In addition, we demonstrate that the auxiliary information is directly applicable to augmenting the neural IR model for increased proportional performance correlating with respect to the NLP probe.

2 RELATED WORK

A number of recent papers have focused on the development of techniques to discover the internal representations of neural networks. Alain and Bengio [2] use a set of linear classifiers to capture the current level of information within the hidden layers of a network. After training a deep neural network for image classification, the weights of the network are frozen and the hidden layers become inputs for their respective linear classifiers. Thus each classifier is trained to predict the label of the example with access to only an intermediate representation.

Zintgraf et al. [36] approach the investigation of neural networks from a different perspective by using prediction difference analysis. Rather than investigating the representation at intermediate layers, they mask a portion of the input and measure the difference in total activation and classification probability. The masks are created by conditioning an area of pixels on the surrounding area. This results in down-weighting easy to predict pixels which can be viewed as redundant and allows the visualization of rare pixels that have the greatest impact on the class score of the image. This allows for a rich exploration when using the mask due to the fully labeled pixel data. However, this same property makes the direct adaptation of their method to text a challenge.

In the field of computer vision, where visualization is much more intuitive for human eyes, there are two main approaches to understanding intermediate layers. The first is similar to Alain and Bengio's [2] work involving intermediate probes. However, these probes attempt to reconstruct the initial input from a convolutional neural network rather than classifying it. The motivation lies in the belief that the pixels of an image that are able to be reconstructed reflect the information contained in a hidden representation. The second approach involves a top down perspective where one can either use gradient updates from the loss function to determine which pixels are most important or deconvolving, where a new supervised model is trained to sequentially project higher layers in the original network to lower ones. However, both of theses approaches require some adaptation for text due to the less intuitive visual representation.

Li et al. [14] utilize this past work and examine one feature at a time by directly analyzing layer activations and using first order Taylor expansions to measure the importance of specific words on the output of the network, referred to as saliency. They explore the impact of individual tokens on a sentiment classification task via LSTM models. Kaparthy et al. [11] provide a more comprehensive review on recurrent character level language models. They use LSTM activations to interpret individual cells and identify long range range dependencies in text.

Adi et al. [1] approach the representation of text by using prediction tasks to characterize what information is being captured. They use a variety of sentence encoding methods and predict the efficacy of these final representations in representing the original sentence's word length, unique word content, and word order. While the final sentence encodings are evaluated, no investigation in the interior of the networks is conducted. Belinkov et al. [3] expand Alain and Bengio's [2] work in the neural machine translation area by inserting classifiers into a neural machine translation model to examine the impact of morphology. While similar to this work in regard to both using Alain and Bengio's methodology, Belinkov's analysis was focused solely on the connection to translation, while this work examines the information loss specific to information retrieval on an arbitrary auxiliary NLP task.

In the specific realm of IR, Palangi et al. [23] examine the hidden activations of long short term memory (LSTM) networks on query-sentence pairs at the word level. They identify key query words within the LSTM activations.

3 METHOD

We investigate the information captured by an IR answer passage deep neural model by utilizing the intermediate representations of deep neural networks. The foundation of this work comes from the data processing inequality [8], where given a markov chain of successive representations, $X \rightarrow Y \rightarrow Z$, then

$$I(X;Z) \leq I(X;Y)$$

where $I(X;Y)$ is the mutual information function. Tishby and Zaslvsky [30] show that layered neural networks form a Markov chain of representations. Thus from the perspective of deep neural models, each transformation at best can preserve the information from an earlier layer and the most information available at any point in a neural model is contained in the lower layers. The advantage of additional layers is not to add information, but to identify the salient information with respect to the target, and transform the representation into a linearly separable space.

We apply this inequality to identify what information is discarded by a passage retrieval model from the perspective of NLP tasks. Specifically, after training an IR model for answer passage retrieval, we freeze the weights and pass it over POS, NER, sentiment, and textual entailment datasets. Each hidden state of the IR model then goes into separate neural networks, or probes, to predict the input's true NLP label. The efficacy of the probes' ability to learn and predict the various NLP tasks illustrates what linguistic and semantic properties are being discarded by the network from the base embeddings.

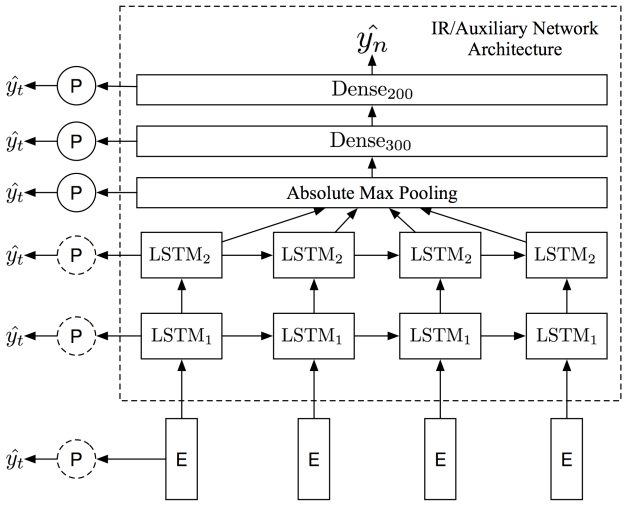

Figure 1: An overview of the probe (P) insertion model. The main LSTM model attempts to evaluate the input for its main task in embedding (E) format, $\hat{y}_n$ while the probes use each layer's intermediate representation to predict an auxiliary task $\hat{y}_t$.

3.1 Model

In this paper, we implement a multilayer LSTM network used by Graves et al. [10] that feeds into a series of dense feed forward layers (Figure 1). This has been shown to be a critical component of state of the art NLP and answer passage retrieval systems [18, 29]. The model consists of two LSTM layers where the output of the second LSTM layer is absolute max pooled. It is then passed to two feed forward dense layers and outputs to a K dimensional final layer. The final layer is a dense layer with the number of nodes equal to the number of classes where a softmax function is taken to create a probability distribution over the labels. Additional hyperparameters are provided in Table 1. We use a dropout rate of 0.2 over both LSTM outputs during training of the main network for all tasks as it has been shown to improve the generalizability of deep neural models to unseen data [34, 35].

Table 1: Hyperparameters of Main and Probe models with K as the final classes for a task

Main Model		
Layer	**Dimension**	**Activation**
LSTM 1	512	internal: sigmoid, output: *tanh*
LSTM 2	512	internal: sigmoid, output: *tanh*
Dense	300	ReLU
Dense	200	ReLU
Dense	K	-
Probe		
Layer	**Dimension**	**Activation**
Dense	300	ReLU
Dense	200	ReLU
Dense	K	-

Auxiliary Networks In order to provide an approximate upper bound for the internal representation and information contained in the IR network described in the previous section, we train additional LSTM networks identical to the IR network dedicated to each auxiliary task. The sole difference between these auxiliary LSTM models and the IR model is the training data and dimension of the final dense layer. The same hyperparameters and training methods were used across all LSTM networks. The difference in probe performance between those inserted in the IR model and those in the auxiliary network can then be viewed as the discarded information with respect to the auxiliary signal.

3.2 Probes

We use small multilayer perceptions, which accept as input each intermediate layer of the main LSTM networks. These probes are trained to predict target labels of the input using only the current hidden layer of the network they are monitoring. Changing the input of the main IR network to reflect an auxiliary task allows the probes to effectively become a measurement tool in how much information the main network is retaining with respect to these auxiliary labels. An illustrative representation of the setup is shown in Figure 1. Any task that requires individual labeling of tokens, denoted by the dashed probe symbol P, are fed the LSTM and embedding sequences in temporal order. Text classification tasks receive (1) a max pooled representation across time of the recurrent layers, (2) a sum of embeddings or (3) direct output from a dense layer as shown below,

$$x_i = \underset{t}{\text{argmax}}(|\mathbf{h}_{i,t}|) \tag{1}$$

$$\mathbf{x} = \sum_{w \in S} \text{Embedding}(w) \tag{2}$$

$$\mathbf{x} = \sigma(W\mathbf{h}_{l-1} + b) \tag{3}$$

where $\mathbf{x}$ is the vector input into the probe, $\mathbf{h}_{i,t}$ is the hidden LSTM layer at dimension i at time t, w represents the words contained in the sample S, and $\mathbf{h}_{l-1}$ is input into the dense layer of the main network and the output is passed to the probe.

3.3 Tasks

We also evaluate the vocabulary overlap between the IR collection and the auxiliary task to ensure that the majority of the new input into the IR network has been seen during training. As shown in Table 3, there is a significant overlap between IR training and task evaluation vocabulary

Core Task: Answer Passage Retrieval: The core IR task being studied is answer passage retrieval. As mentioned, this task represents a unique challenge when compared to ad-hoc retrieval and factoid QA. While factoid retrieval often encounters questions such as "*When did James Dean Die*" or "*How high is Everest?*" that require only one or two tokens to successfully fulfill the information need of the query, passage retrieval requires information that spans multiple sentences. This integral difference results in state of the art factoid QA networks failing to beat standard *tf-idf* baselines on answer passage retrieval tasks [7].

Auxiliary Task: Part of Speech Tagging: Part of speech (POS) tagging is the task of labeling each word with its syntactic part of speech, e.g. noun, verb, adjective, based on its use in a sentence.

As shown in past work by Bjerva et al. [4], networks trained on semantic tagging tasks independently capture part of speech information. As passage retrieval requires semantic processing to bridge the information across sentences, we investigate the extent to which an answer passage neural model also captures POS tags.

Auxiliary Task: Named Entity Tagging: While related to POS tagging, named entity recognition (NER) requires higher level features which often consist of POS information, whether latent or explicit, due to the dependencies across a sentence and additional information required for accurate entity tagging [13, 15]. We evaluate a NER auxiliary task to see if the core answer passage network dedicates some of its parameters to capture information pertaining to named entities. Recent work in deep neural QA [26] have shown that adding named entity overlap between question and answer significantly improves performance with respect to IR metrics. The increase in IR metrics suggests that entity information plays an integral role for modeling relevance.

Auxiliary Task: Sentiment Classification: At a significantly higher level task compared to POS tagging and NER due to the need to process and compress an entire sequence, we implement a sentiment classification task. Here, the objective is correctly identify whether a sentence denotes a positive or negative view of the topic. Li et al.'s [14] work in visualizing LSTM networks for sentiment classification provides insight on what features are important to predict sentiment. The most critical components are the ability to capture local context around a word, recognize negation, qualitative adjectives and key verbs.

Auxiliary Task: Textual Entailment: We use a textual entailment task to evaluate whether information retrieval at the passage level could be viewed as whether the query provides evidence for a passage to be considered relevant. The goal of this task is to determine whether two sentences (1) are contradicting each other, (2) are unrelated, or (3) that the first sentence (the evidence) entails the hypothesis. The performance of the probes on the core IR model will help disentangle the semantic information related to entailment over that which relates a query to its relevant passage. As each example is an ordered pair of sentences, we concatenate the evidence-hypothesis sentences the same way as query-passage pairs for the answer passage retrieval task. The evidence serves as the query and the passage represents the hypothesis. The auxiliary network and probes for this task have a three node dense final layer to classify entailment, contradiction, and neutral classes.

3.4 Multitask Inspection

As information retained in each layer has some benefit towards determining relevance, we examine the impact of explicitly reinforcing this signal through a multitask environment using a similar neural structure as Long and Wang [16] where gradients are passed through task specific sub networks into larger main model. Thus the probe remains task dependent while the layer of the IR network it connects to, and those below it, become shared layers for the multitask objective. This approach retains the probe inspection method while simultaneously adopting a competitive neural multitask framework. As the IR collections do not have gold NLP labels for training, we use the trained auxiliary NLP networks to create pseudo labels for training.

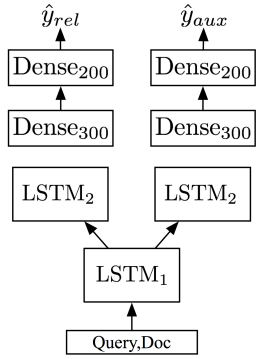

Figure 2: Simplified representation of multitask architecture with LSTM$_1$ acting as shared layer.

The structure of the multitask architecture consists of the main model hyperparameters described in Table 1. The corresponding task specific substructures are mirrored. Thus if the shared layer is LSTM 1, then LSTM 2 and the subsequent feedforward hyperparameters are used for both tasks with no weight sharing. A depiction of this setup is exemplified in Figure2. The multitask model is optimized via the joint loss function

$$L = L_{aux} + L_{IR}$$

where L_{aux} and L_{IR} are the respective loss functions used for single task training discussed in the following section.

3.5 Training

We use Adam for optimizing both the main models as well as the probes with a cross-entropy loss function and a learning rate of 10^{-3} , which provides a robust value for training [12]. Each main model was trained via PyTorch[1] over a 80-10-10 train, development, and test partition and was stopped after the best validation loss did not improve for four epochs as a form of early stopping. Each probe was trained, validated, and tested on the same data to measure the amount of information captured by the main model rather than the probe's ability to generalize.

Input into the IR network is done in a similar manner to past works [7, 31] by concatenating question and passage text with an end-of-sentence (EOS) token as shown below.

$$< q_1, \ldots, q_n > + < \text{EOS} > + < a_1, \ldots, a_m >$$

This allows for query passage interaction while still being easily adaptable to processing input from auxiliary tasks. In the case of the NLP tasks that do not have text pairs to partition with <EOS>, we feed the text in directly to simulate the query stage of an IR task, and then we train another set of probes on samples where <EOS> is prepended to the same the sample. The IR network views this as an empty query and a candidate passage, which enables us to identify how captured information differs for the same text as query and passage in the IR main network.

All tokens are expressed as GLOVE 300D embeddings [2] [24]. In order to provide a consistent text representation across all tasks, we

[1] https://github.com/pytorch/pytorch
[2] http://nlp.stanford.edu/data/glove.840B.300d.zip

do not update the initial embeddings during training at any point. This represents a common baseline across all models.

3.5.1 Datasets. **Answer Passage Retrieval:** To investigate what information is pertinent to an answer passage network for determining the relevance of a candidate document, we use Yahoo's Webscope L4 high quality "Manner" collection [28] and a noisier nfl6 collection derived from Yahoo's Webscope L6 [7] referred to as the CQA collection. The answer passages within these collections are significantly longer than those found in conventional QA. The average length of the L4 and nfl6 answer passages are 92 and 60 words respectively, while WikiQA [32] sentences have an average length of 25 words. The final dense layer of this task's main model consists of a single node and trained via a binary cross entropy loss. We apply the following auxiliary methods below not just to the CQA dataset, but to WikiQA to provide insight into why certain models dramatically suffer a significant loss in performance when moving from sentence QA to passage retrieval.

Part of Speech Tagging: The collection used for evaluating this auxiliary task is the Wall Street Journal set from Penn Treebank III [20]. As mentioned, POS probes are inserted only into the temporal (LSTM and embedding) layers of the core network. The POS auxiliary main network was trained over 46 POS using the standard train (0-20), validation (21, 22), and test (23, 24) splits as seen in past work [21].

Named Entity Recognition: We use the CoNLL-2003 NER for training and evaluation [25] and use *MISC, LOC, ORG, PERS, O* tags over the standard BIO annotation (*Begin, Inside, Outside*). This was done to investigate whether the IR main model is able to identify and differentiate among the classes over the more detailed task of determining whether a token is the beginning of a phrase or inside it.

Sentiment Classification: Like the past work [14], we use the Internet Movie Database (IMDB) review collection [19] where a movie review is either positive or negative with 25k samples for each label. We use binary cross entropy to evaluate this task.

Textual Entailment: We use Stanford's Natural Language Inference (SNLI) corpus [5]. This entailment set is a collection of 570k human-written English sentence pairs with labels of *entailment* (183,416), *contradiction* (183,187), and *neutral* (182,764). We discard the 785 samples that do not fall under one of these three labels. Each sample is an ordered pair of sentences, one that serves as the evidence and the following that is a hypothesis. The auxiliary network and probes for this task have a three node dense final layer to classify entailment, contradiction, and neutral classes.

4 RESULTS AND DISCUSSION

In this section, we show the performance of the probes using the answer passage network's internal representation.

As shown in Table 4, there is a steady decline in information loss as the initial embeddings flow up through the layers. Following work found in computer vision [33] where each layer captures increasingly abstract representations, the answer passage model also reflects this tendency. Lower level POS and NER information is captured consistently in the first LSTM layer and discarded in the upper layers, while the abstract entailment information persists into the model's upper layers, even sharing some of the transformations needed to determine passage relevance. This reinforces the analysis done by

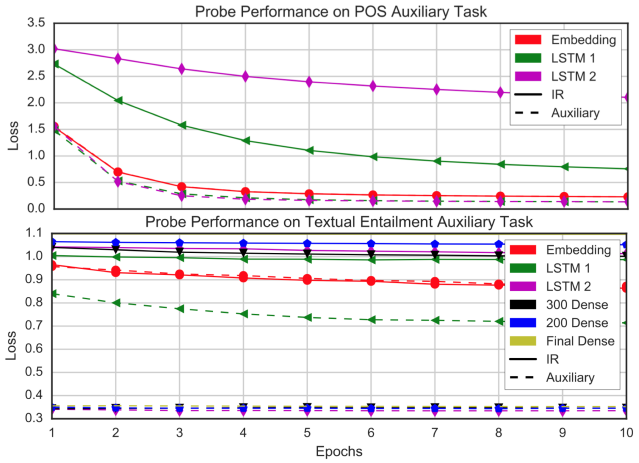

Figure 3: Performance of probes over each layer on all auxiliary tasks as queries. IR represents the probes inserted into the answer passage network and Auxiliary represents probes inserted into the identical network trained for the auxiliary task.

Søogaard and Goldberg [27], where they had greater success with a neural architecture that supervised POS information at the lower layers for multitask learning. Lastly, we show that two seemingly similar IR tasks that are considered closely related have significantly different information needs.

Part of Speech Tagging: The performance during training (Figure 3) highlights the large degree of stratification of information that the IR network is undergoing when learning relevance. Reflected in the loss function, the initial embeddings retain the most POS information while the subsequent LSTM layers suffer a decrease in F1 within the IR model. However, moving from the first to second LSTM layer in the core IR model receives a much greater 50% loss in performance. This large degradation suggests that as a somewhat low level feature, POS information is still captured in the hidden representation of the higher LSTM layer albeit in a much weaker representation. The slower slope of the loss function on $LSTM_2$ and significantly degraded F1 score, combined with Palangi et al.'s [23] work on LSTM networks learning a rough topical model, suggests that the probe is learning to recognize more abstract topical representations and mapping them to POS labels. The difference in performance across query and passage representations indicates that the IR network attends to POS information equally.

Named Entity Recognition: Closely related to POS tagging, we analyze the probes' performance on the NER auxiliary task. Probe performance on the auxiliary network shows a greater need for capturing abstract and contextual information than POS tagging due to the separation in performance of the embeddings and LSTM layers in both loss and F1 over epochs on the auxiliary NER network.

Examining the probes within the IR network reinforces the evidence that the second LSTM layer is learning a more topical representation. However, as the second LSTM layer discards a significant amount of POS information, the sustained performance on the NER task across LSTM layers suggests that the IR model uses named

Collection: nfl6
Q: Why do teachers go abroad? A: Many reasons: First, because its really 'freakin' cool to go abroad and sample different cultures and languages. Two: Alot of schools abroad offer great packages, like free room and board while your teaching there and some even offer to pay for your current or impending degree and any travel expenses. Three:Some american schools offer exchange programs for teachers, so that teachers can go abroad to experiment and learn about different and alternative teaching styles. Normally teachers are compensated for it, kinda like a sabbatical leave
Collection: Webscope L4
Q: How can I safely open a geode? A: One way to open a geode – to reveal the crystals – is with a chisel and hammer. Score the geode completely around the outside where you want it to crack – usually in two equal halves. Keep going until it cracks and breaks apart. This will almost always work and won't damage the crystals.

Table 2: An example query answer pair from the two passage collections being considered.

Table 3: Vocabulary overlap measured by $\frac{A_i \cap B_j}{B_j}$ between auxiliary collections (B_j) and the two IR collections (A_i).

Task	L4+nfl6		WikiQA	
	Unique	Total	Unique	Total
CoNLL 2003	.595	.649	.424	.565
PTB II	.769	.815	.584	.673
IMDB	.293	.972	.110	.916
SNLI	.720	.994	.383	.952

entities for passage length relevance judgements either through explicit capturing at the cell level, or in a latent representation in the hidden layer independent of POS information. The F1 drop when processing the same samples from a passage perspective indicates that the IR network focuses on capturing more information related to named entities when processing text at the query stage. However, the drop in performance could also be due to the lack of relevant query text priming the network to focus on named entity information.

Sentiment Classification: Moving to a more abstract task requiring an entire sentence, probes trained to label sentiment result in a significantly different outcome than NER and POS. Each layer remains a close neighbor to its subsequent one when viewed from the probes' perspectives. The small decrease in sentiment classification performance, accompanied with thet large loss of POS information suggests that some form of more abstract sentiment information is captured in each layer. Furthermore, this can be assumed to be explicitly modeled within the cell due to the almost zero slope of the loss during training that mirrors the entailment task's rate (Figure 3). However, while sentiment information is used for establishing relevance, there is no signal present in the actual relevance label, as shown by Dense_y's result of 0.49 and the random model receiving a 0.50 accuracy score. In addition, the IR network does not seem to process sentiment information differently across query and passage text as seen by the relatively stable performance in Table 4.

Entailment: Confirming the results in the previous subsections, the most abstract task of capturing entailment suffers the least across layers. Additionally, contrary to the other auxiliary tasks, higher layers significantly outperform the lower ones as seen in the difference between LSTM_1 and Dense_{300} in Table 4. Accounting for the accuracy across other tasks, the increased performance of the third layer, Dense_{300}, suggests that the transformations used to determine relevance at this point also act to move entailment classes into a more linearly separable space.

Lastly, the performance of the probe on the relevance score, Dense_y, shows that the relevance of a query passage pair has some information with respect to logical entailment. We expand this insight and investigate individual label performance as shown in Table 5. The individual label evaluations show that each of the three classes requires unique information. In addition, the relevance model retains information for detecting entailment, while information for neutral and contradictory labels is iteratively discarded at each layer. The following dip in performance in Dense_{200} indicates that the upper layers put less emphasis on entailment. Finally, looking at the relation between the scalar relevance value, Dense_y, and the individual label metrics shows that positive entailment information is related to the relevance of a passage, although non relevant documents provide no indication that the query and passage pair do not contain some type of entailment.

4.1 Multitask Inspection

Examining the impact of the auxiliary loss signal for IR, the same trend as seen in Table 4 occurs in Figure 4, where the layer that captures the most information with respect to the auxiliary task is also the most effective layer within the multitask environment for retrieval. Of particular interest is the NER performance on WikiQA. This task significantly improves performance when using LSTM_1 as the shared layer, and subsequently suffers the greatest performance decrease across all tasks when moving upward. This suggests that for retrieval on this collection, the use of named entities within a neural model is heavily biased towards the first layer, not only from an information perspective, but also from a performance view as well. Lastly, following the trend in Table 5, the multitask model over CQA benefits from using LSTM_2 as the shared layer, where the most information used for entailment is captured. This also demonstrates that the optimal shared multitask layer for retrieval is not the lowest by default, as it is for POS and NER auxiliary tasks.

4.2 Dataset Comparison

WikiQA vs CQA: As mentioned in section 3.5, we perform the same NLP auxiliary analysis on an additional factoid QA dataset. Shown in Table 4, there exists a consistent decline in performance from the lower to upper layers. However, due to the greater amount of factoid type queries, the WikiQA model retains more information with respect to NER and POS information at the cost of reduced performance on sentiment and entailment tasks. Not only does the WikiQA model perform worse than the CQA model on these tasks, but the hidden transformations used for determining relevance fail to provide any assistance regarding separating entailment unlike the CQA model. While the WikiQA dataset shares significantly

Table 4: F1 score for NER and POS, and Accuracy for Sentiment and Entailment tasks of each layer of the IR network over auxiliary NLP tasks with input treated as the query. *Aux* in the second column represents the probes inserted into an identical LSTM network trained directly on the auxiliary task. Parenthesis indicates performance difference when placing <EOS> prior to sample input, and bold shows best layer on each task.

CQA								
Layer	**NER**		**POS**		**Sentiment**		**Entailment**	
	IR	**Aux**	**IR**	**Aux**	**IR**	**Aux**	**IR**	**Aux**
Random	.200		.022		.500		.333	
Embedding	.963		.917		.844		.590	
$LSTM_1$	**.927(-.001)**	.987	**.751(-.001)**	.951	**.721(-.004)**	.900	.522(+.036)	.715
$LSTM_2$	.810(-.002)	**.987**	.305(-.002)	**.954**	.666(-.001)	.900	.518(+.040)	.873
$Dense_{300}$	-	-	-	-	.689(-.005)	.926	**.527(+.039)**	.877
$Dense_{200}$	-	-	-	-	.668(-.007)	.932	.454(+.031)	.881
$Dense_y$	-	-	-	-	.498(+.006)	**.934**	.366(-.008)	**.885**
WikiQA								
Layer	**NER**		**POS**		**Sentiment**		**Entailment**	
	IR	**Aux**	**IR**	**Aux**	**IR**	**Aux**	**IR**	**Aux**
Random	.200		.022		.500		.333	
Embedding	.963		.917		.844		.590	
$LSTM_1$	**.934(-.001)**	.987	**.794(-.000)**	.951	**.638(+.001)**	.900	**.464(-.010)**	.715
$LSTM_2$	.845(-.001)	**.987**	.386(+.051)	**.954**	.593(-.001)	.900	.425(+.020)	.873
$Dense_{300}$	-	-	-	-	.572(-.003)	.926	.400(+.018)	.877
$Dense_{200}$	-	-	-	-	.557(-.001)	.932	.377(+.021)	.881
$Dense_y$	-	-	-	-	.503(+.003)	**.934**	.355(-.002)	**.885**

Table 5: Per label accuracy performance over SNLI entailment collection on CQA model. E, N, C represent the classes *Entailment, Neutral*, and *Contradiction*.

Layer	**E**	**N**	**C**
Random	.333	.333	.333
Embedding	.593	.531	.624
$LSTM_1$	.569	.466	**.517**
$LSTM_2$	.596	.473	.470
$Dense_{300}$	**.610**	**.476**	.480
$Dense_{200}$	.529	.403	.422
$Dense_y$	.424	.372	.296

less vocabulary overlap than the CQA collection as seen in Table 3, examining the impact of missing vocabulary on the incorrectly classified auxiliary samples reveals a Pearson's correlation of 0.194, and restricting the CQA collection to the same as the WikiQA training set, 12,888 random samples, does not significantly reduce performance on the auxiliary tasks. This provides insight in why some past models that perform successfully on shorter QA tasks struggle on passage retrieval [6].

5 CONCLUSION

The proposed probe based NLP view of retrieval models identifies key information used by these models that has not been identified in past work. While we demonstrate the well established result that increasingly abstract features are learned within an IR network, we show in detail that the same information is used in a significantly different manner across collection types. Furthermore, this information can directly improve performance via adding additional structures to an IR model through multitask learning. While preliminary, this result indicates that using rigid subtasks to represent retrieval features is a promising avenue for future work.

6 ACKNOWLEDGMENTS

We thank Emma Strubell, Katie Keith, Rajarshi Das, Hamed Zamani, and John Foley for their constructive comments and insights.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via AFRL contact #FA8650-17-C-9116 under subcontract #94671240 from the University of Southern California. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

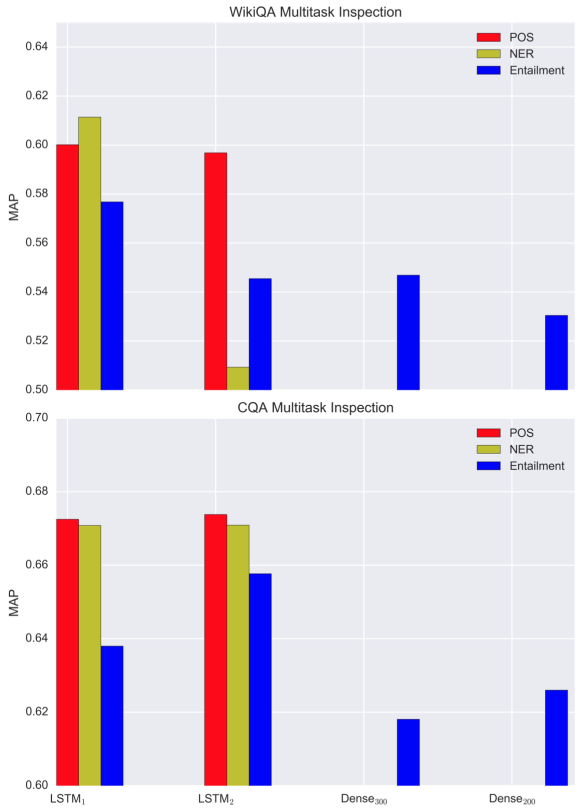

Figure 4: Per-layer performance of NER, POS, and Entailment tasks measured by MAP on WikiQA and CQA collections.

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *CoRR* abs/1608.04207 (2016). arXiv:1608.04207 http://arxiv.org/abs/1608.04207

[2] Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *CoRR* abs/1610.01644 (2016). arXiv:1610.01644 http://arxiv.org/abs/1610.01644

[3] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do Neural Machine Translation Models Learn about Morphology?. In *ACL*. 861–872. https://doi.org/10.18653/v1/P17-1080

[4] Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic Tagging with Deep Residual Networks. In *COLING 2016, Technical Papers, December 11-16, 2016, Osaka, Japan*. 3531–3541.

[5] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *MNLP*. ACL.

[6] Daniel Cohen, Qingyao Ai, and W. Bruce Croft. 2016. Adaptability of Neural Networks on Varying Granularity IR Tasks. In *SIGIR Neu-IR Workshop*. Pisa, Italy.

[7] Daniel Cohen and W. Bruce Croft. 2016. End to End Long Short Term Memory Networks for Non-Factoid Question Answering. In *ICTIR*. Newark, DE, USA.

[8] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

[9] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. In *IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 3460–3468. http://ijcai.org/Abstract/15/487

[10] Alex Graves, Abdel-Rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*. IEEE, 6645–6649.

[11] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and Understanding Recurrent Networks. *CoRR* abs/1506.02078 (2015).

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980

[13] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *HLT-NAACL*.

[14] Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016. Visualizing and Understanding Neural Models in NLP. In *NAACL HLT 2016, San Diego California, USA, June 12-17, 2016*. 681–691.

[15] Jiwei Li and Dan Jurafsky. 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding?. In *EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 1722–1732.

[16] Mingsheng Long and Jianmin Wang. 2015. Learning Multiple Tasks with Deep Relationship Networks. *CoRR* abs/1506.02117 (2015). arXiv:1506.02117 http://arxiv.org/abs/1506.02117

[17] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*. The Association for Computational Linguistics, 1412–1421.

[18] Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL 2016, August 7-12, 2016, Berlin, Germany*. http://aclweb.org/anthology/P/P16/P16-1101.pdf

[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *ACL*. Association for Computational Linguistics, Portland, Oregon, USA, 142–150.

[20] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *HLT*. Association for Computational Linguistics, Stroudsburg, PA, USA, 114–119. https://doi.org/10.3115/1075812.1075835

[21] TomÃąÅą Mikolov, Ilya Sutskever, Anoop Deoras, Le Hai Son, Stefan Kombrink, and JaÅĹ Cernock. 2010. Subword Language Modeling with Neural Networks. *preprint* (2010).

[22] Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. Improving Document Ranking with Dual Word Embeddings. In *WWW*. Republic and Canton of Geneva, Switzerland, 83–84. https://doi.org/10.1145/2872518.2889361

[23] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE/ACM Trans. Audio, Speech & Language Processing* 24, 4 (2016), 694–707. https://doi.org/10.1109/TASLP.2016.2520371

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[25] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. 142–147.

[26] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR (SIGIR '15)*. ACM, New York, NY, USA, 373–382. https://doi.org/10.1145/2766462.2767738

[27] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *ACL (2)*. The Association for Computer Linguistics.

[28] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *ACL:HLT*. 719–727.

[29] Ming Tan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. In *ACL (1)*. The Association for Computer Linguistics.

[30] Naftali Tishby and Noga Zaslavsky. 2015. Deep Learning and the Information Bottleneck Principle. *CoRR* abs/1503.02406 (2015).

[31] Di Wang and Eric Nyberg. 2015. A Recurrent Neural Network based Answer Ranking Model for Web Question Answering. In *WebQA Workshop, SIGIR '15, Santiago, Chile*.

[32] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 2013–2018.

[33] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015). https://arxiv.org/abs/1506.06579

[34] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent Neural Network Regularization. *CoRR* abs/1409.2329 (2014). arXiv:1409.2329 http://arxiv.org/abs/1409.2329

[35] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. *ICLR* (2017). arXiv:1611.03530 http://arxiv.org/abs/1611.03530

[36] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. *CoRR* abs/1702.04595 (2017). arXiv:1702.04595 http://arxiv.org/abs/1702.04595