

HARE: a Flexible Highlighting Annotator for Ranking and Exploration

Denis Newman-Griffis^{†,‡} and Eric Fosler-Lussier[†]

[†]Dept. of Computer Science and Engineering, The Ohio State University, Columbus, OH

[‡]Rehabilitation Medicine Dept., Clinical Center, National Institutes of Health, Bethesda, MD

{newman-griffis.1, fosler-lussier.1}@osu.edu

Abstract

Exploration and analysis of potential data sources is a significant challenge in the application of NLP techniques to novel information domains. We describe HARE, a system for highlighting relevant information in document collections to support ranking and triage, which provides tools for post-processing and qualitative analysis for model development and tuning. We apply HARE to the use case of narrative descriptions of mobility information in clinical data, and demonstrate its utility in comparing candidate embedding features. We provide a web-based interface for annotation visualization and document ranking, with a modular backend to support interoperability with existing annotation tools.

1 Introduction

As natural language processing techniques become useful for an increasing number of new information domains, it is not always clear how best to identify information of interest, or to evaluate the output of automatic annotation tools. This can be especially challenging when target data in the form of long strings or narratives of complex structure, e.g., in financial data (Fisher et al., 2016) or clinical data (Rosenbloom et al., 2011).

We introduce HARE, a Highlighting Annotator for Ranking and Exploration. HARE includes two main components: a workflow for supervised training of automated token-wise relevancy taggers, and a web-based interface for visualizing and analyzing automated tagging output. It is intended to serve two main purposes: (1) triage of documents when analyzing new corpora for the presence of relevant information, and (2) interactive analysis, post-processing, and comparison of output from different annotation systems.

In this paper, we demonstrate an application of HARE to information about individuals' mo-

bility status, an important aspect of functioning concerned with changing body position or location. This is a relatively new type of health-related narrative information with largely uncharacterized linguistic structure, and high relevance to overall health outcomes and work disability programs. In experiments on a corpus of 400 clinical records, we show that with minimal tuning, our tagger is able to produce a high-quality ranking of documents based on their relevance to mobility, and to capture mobility-likely document segments with high fidelity. We further demonstrate the use of post-processing and qualitative analytic components of our system to compare the impact of different feature sets and tune processing settings to improve relevance tagging quality.

2 Related work

Corpus annotation tools are plentiful in NLP research: brat (Stenetorp et al., 2012) and Knowtator (Ogren, 2006) being two heavily used examples among many. However, the primary purpose of these tools is to streamline *manual* annotation by experts, and to support review and revision of manual annotations. Some tools, including brat, support automated pre-annotation, but analysis of these annotations and corpus exploration is not commonly included. Other tools, such as SciKnowMine,¹ use automated techniques for triage, but for routing to experts for curation rather than ranking and model analysis. Document ranking and search engines such as Apache Lucene,² by contrast, can be overly fully-featured for early-stage analysis of new datasets, and do not directly offer tools for annotation and post-processing.

Early efforts towards extracting mobility information have illustrated that it is often syntactically

¹<https://www.isi.edu/projects/sciknowmine/overview>

²<https://lucene.apache.org/>

	SpaCy	WordPiece
Num documents		400
Avg tokens per doc	537	655
Avg mobility tokens per doc	97	112
Avg mobility segments per doc		9.2

Table 1: Statistics for dataset of mobility information, using SpaCy and WordPiece tokenization.

and semantically complex, and difficult to extract reliably (Newman-Griffis and Zirikly, 2018; Newman-Griffis et al., 2019). Some characterization of mobility-related terms has been performed as part of larger work on functioning (Skube et al., 2018), but a lack of standardized terminologies limits the utility of vocabulary-driven clinical NLP tools such as CLAMP (Soysal et al., 2018) or cTAKES (Savova et al., 2010). Thus, it forms a useful test case for HARE.

3 System Description

Our system has three stages for analyzing document sets, illustrated in Figure 1. First, data annotated by experts for token relevance can be used to train relevance tagging models, and trained models can be applied to produce relevance scores on new documents (Section 3.1). Second, we provide configurable post-processing tools for cleaning and smoothing relevance scores (Section 3.2). Finally, our system includes interfaces for reviewing detailed relevance output, ranking documents by their relevance to the target criterion, and analyzing qualitative outcomes of relevance scoring output (Sections 3.3-3.5); all of these interfaces allow interactive re-configuration of post-processing settings and switching between output relevance scores from different models for comparison.

For our experiments on mobility information, we use an extended version of the dataset described by Thieu et al. (2017), which consists of 400 English-language Physical Therapy initial assessment and reassessment notes from the Rehabilitation Medicine Department of the NIH Clinical Center. These text documents have been annotated at the token level for descriptions and assessments of patient mobility status. Further information on this dataset is given in Table 1. We use ten-fold cross validation for our experiments, splitting into folds at the document level.

3.1 Relevance tagging workflow

All hyperparameters discussed in this section were tuned on held-out development data in cross-

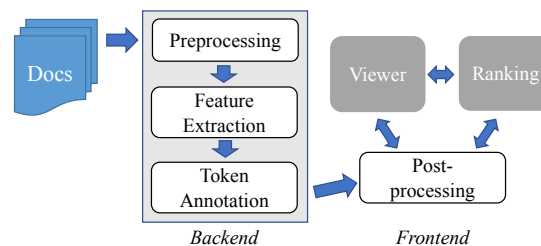


Figure 1: HARE workflow for working with a set of documents; outlined boxes indicate automated components, and gray boxes signify user interfaces.

validation experiments. We report the best settings here, and provide full comparison of hyperparameter settings in Appendix A.

3.1.1 Preprocessing

Different domains exhibit different patterns in token and sentence structure that affect preprocessing. In clinical text, tokenization is not a consensus issue, and a variety of different tokenizers are used regularly (Savova et al., 2010; Soysal et al., 2018). As mobility information is relatively unexplored, we relied on general-purpose tokenization with spaCy (Honnibal and Montani, 2017) as our default tokenizer, and WordPiece (Wu et al., 2016) for experiments using BERT. We did not apply sentence segmentation, as clinical toolkits often produced short segments that interrupted mobility information in our experiments.

3.1.2 Feature extraction

Our system supports feature extraction for individual tokens in input documents using both static and contextualized word embeddings.

Static embeddings Using static (i.e., non-contextualized) embeddings, we calculate input features for each token as the mean embedding of the token and 10 words on each side (truncated at sentence/line breaks). We used FastText (Bojanowski et al., 2017) embeddings trained on a 10-year collection of physical and occupational therapy records from the NIH Clinical Center.

ELMo (Peters et al., 2018) ELMo features are calculated for each token by taking the hidden states of the two bLSTM layers and the token layer, multiplying each vector by learned weights, and summing to produce a final embedding. Combination weights are trained jointly with the token annotation model. We used a 1024-dimensional ELMo model pretrained on PubMed data³ for our

³<https://allennlp.org/elmo>

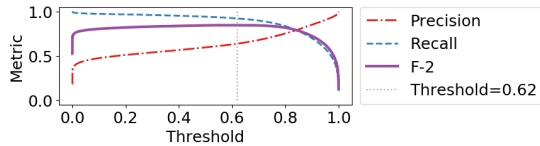


Figure 2: Precision, recall, and F-2 when varying binarization threshold from 0 to 1, using ELMo embeddings. The threshold corresponding to the best F-2 is marked with a dotted vertical line.

mobility experiments.

BERT (Devlin et al., 2019) For BERT features, we take the hidden states of the final k layers of the model; as with ELMo embeddings, these outputs are then multiplied by a learned weight vector, and the weighted layers are summed to create the final embedding vectors.⁴ We used the 768-dimensional clinicalBERT (Alsentzer et al., 2019) model⁵ in our experiments, extracting features from the last 3 layers.

3.1.3 Automated token-level annotation

We model the annotation process of assigning a relevance score for each token using a feed-forward deep neural network that takes embedding features as input and produces a binomial softmax distribution as output. For mobility information, we used a DNN with three 300-dimensional hidden layers, relu activation, and 60% dropout.

As shown in Table 1, our mobility dataset is considerably imbalanced between relevant and irrelevant tokens. To adjust for this balance, for each epoch of training, we used all of the relevant tokens in the training documents, and sampled irrelevant tokens at a 75% ratio to produce a more balanced training set; negative points were re-sampled at each epoch. As token predictions are conditionally independent of one another given the embedding features, we did not maintain any sequence in the samples drawn. Relevant samples were weighted at a ratio of 2:1 during training.

After each epoch, we evaluate the model on all tokens in a held-out 10% of the documents, and calculate F-2 score (preferring recall over precision) using 0.5 as the binarization threshold of model output. We use an early stopping threshold of $1e-05$ on this F-2 score, with a patience of 5

⁴Note that as BERT is constrained to use WordPiece tokenization, it may use slightly longer token sequences than the other methods.

⁵<https://github.com/EmilyAlsentzer/clinicalBERT>

treatment difficulty , noting weakness in the hands and lower extremities , reporting more difficulty with performing his daily tasks . Since that time , the patient reports that he has had some

(a) No collapsing

treatment difficulty , noting weakness in the hands and lower extremities , reporting more difficulty with performing his daily tasks . Since that time , the patient reports that he has had some

(b) Collapse one blank

Figure 3: Collapsing adjacent segments illustration.

epochs and a maximum of 50 epochs of training.

3.2 Post-processing methods

Given a set of token-level relevance annotations, HARE provides three post-processing techniques for analyzing and improving annotation results.

Decision thresholding The threshold for binarizing token relevance scores is configurable between 0 and 1, to support more or less conservative interpretation of model output; this is akin to exploring the precision/recall curve. Figure 2 shows precision, recall, and F-2 for different thresholding values from our mobility experiments, using scores from ELMo embeddings.

Collapsing adjacent segments We consider any contiguous sequence of tokens with scores at or above the binarization threshold to be a relevant *segment*. As shown in Figure 3, multiple segments may be interrupted by irrelevant tokens such as punctuation, or by noisy relevance scores falling below the binarization threshold. As multiple adjacent segments may inflate a document’s overall relevance, our system includes a setting to collapse any adjacent segments that are separated by k or fewer tokens into a single segment.

Viterbi smoothing By modeling token-level decisions as conditionally independent of one another given the input features, we avoid assumptions of strict segment bounds, but introduce some noisy output, as shown in Figure 4. To reduce some of this noise, we include an optional smooth-

the L leg and R leg . In addition , patient was unable to perform the unilateral stance on either leg test with eyes closed .

Functional Assessment :

Standardized testing for SBMA . Use of adult myopathy assessment tool

(a) Without smoothing

the L leg and R leg . In addition , patient was unable to perform the unilateral stance on either leg test with eyes closed .

Functional Assessment :

Standardized testing for SBMA . Use of adult myopathy assessment tool

(b) With smoothing

Figure 4: Illustration of Viterbi smoothing.

GSC-GSC_21331_21331.xml

Viewing annotations from: ELMo [Back to document list](#)

Prior level of function : independent in all mobility and community ambulation including elevations and stairs .

Current level of function : Same but with great difficulty . She reports falling on stairs recently , falling in parking lot recently , a few near falls with slipping in bathroom recently . Walking is difficult and painful at both knees .

Patient 's goals : safe ambulation with reduced or removed knee pain .

Precautions : universal .

OBJECTIVE : Patient is alert , oriented x3 , pleasant and cooperative .

Pain : Patient reports pain at 10/10 most of the time in constant capacity . She reports 8/10 at rest at both knees . Her c / o knee pain bilaterally and she uses cold 1 - 2x / daily without relief . Pain is worse at night and with activity . She does not report things that make pain better .

Labeling Settings

Threshold: 0.5

Blanks: 0

Use Viterbi smoothing:

Document Statistics

Num predicted segments: 101

Num true segments: 15

Num predicted tokens: 642

Num true tokens: 203

Token-level accuracy: 53.32

Token-level precision: 25.86

Token-level recall: 81.77

Token-level F-1: 39.29

Token-level F-2: 57.08

Figure 5: Annotation viewer interface.

ing component based on the Viterbi algorithm.

We model the “relevant”/“irrelevant” state sequence discriminatively, using annotation model outputs as state probabilities for each timestep, and calculate the binary transition probability matrix by counting transitions in the training data. We use these estimates to decode the most likely relevance state sequence R for a tokenized line T in an input document, along with the corresponding path probability matrix W , where $W_{j,i}$ denotes the likelihood of being in state j at time i given r_{i-1} and t_i . In order to produce continuous scores for each token, we then backtrace through R and assign score s_i to token t_i as the conditional probability that r_i is “relevant”, given r_{i-1} . Let $Q_{j,i}$ be the likelihood of transitioning from state R_{i-1} to j , conditioned on T_i , as:

$$Q_{j,i} = \frac{W_{j,i}}{W_{R_{i-1},i-1}} \quad (1)$$

The final conditional probability s_i is calculated by normalizing over possible states at time i :

$$s_i = \frac{Q_{1,i}}{Q_{0,i} + Q_{1,i}} \quad (2)$$

These smoothed scores can then be binarized using the configurable decision threshold.

3.3 Annotation viewer

Annotations on any individual document can be viewed using a web-based interface, shown in Figure 5. All tokens with scores at or above the decision threshold are highlighted in yellow, with each contiguous segment shown in a single highlight. Configuration settings for post-processing methods are provided, and update the displayed annotations when changed. On click, each token will display the score assigned to it by the annotation model after post-processing. If the document being viewed is labeled with gold annotations, these are shown in bold red text. Addition-

Viewing annotations from: ELMo [Back to annotation set list](#)

Ranking Statistics

Num documents: 396

Spearman's rho: 74.3

Gold scorer: CountSegmentsAndTokens

Model scorer: CountSegmentsAndTokens

Threshold: 0.5

Blanks: 0

Use Viterbi smoothing:

Order by gold:

Show gold classes:

Class Threshold Background Foreground

Low: 20.0 #7bccc4 #ffff

Some: 6.0 #bae4bc #000000

Little: 1.0 #f09e88 #000000

None: 0.0 #ffffff #000000

Document	Model ranking	Model score	Gold ranking	Gold score
GSC-GSC_21372_21372.xml	1	2111925	1	15454
GSC-GSC_22265_22265.xml	2	21578	4	28226
GSC-GSC_21281_21281.xml	3	21519	276	574
GSC-GSC_22267_22267.xml	4	201404	3	34593
GSC-GSC_21418_21418.xml	5	20488	52	15343
GSC-GSC_21434_21434.xml	6	19367	12	23324
GSC-GSC_21245_21245.xml	7	18310	102	12230
GSC-GSC_21416_21416.xml	8	17360	68	14279
GSC-GSC_21426_21426.xml	9	17343	69	14248
GSC-GSC_21324_21324.xml	10	17283	177	9126
GSC-GSC_22235_22235.xml	11	16490	55	15281
GSC-GSC_22231_22231.xml	12	15479	2	35333
GSC-GSC_21341_21341.xml	13	15404	71	14207

Figure 6: Ranking interface.

ally, document-level summary statistics and evaluation measures, with current post-processing, are displayed next to the annotations.

3.4 Document set ranking

3.4.1 Ranking methods

Relevance scoring methods are highly task-dependent, and may reflect different priorities such as information density or diversity of information returned. In this system, we provide three general-purpose relevance scorers, each of which operates after any post-processing.

Segments+Tokens Documents are scored by multiplying their number of relevant segments by a large constant and adding the number of relevant tokens to break any ties by segment count. As relevant information may be sparse, no normalization by document length is used.

SumScores Documents are scored by summing the continuous relevance scores assigned to all of their tokens. As with the Segments+Tokens scorer, no adjustment is made for document length.

Density Document scores are the ratio of binarized relevant tokens to total number of tokens.

The same scorer can be used to rank gold annotations and model annotations, or different scorers can be chosen. Ranking quality is evaluated using Spearman’s ρ , which ranges from -1 (exact opposite ranking) to +1 (same ranking), with 0 indicating no correlation between rankings. We use Segments+Tokens as default; a comparison of ranking methods is in Appendix B.

3.4.2 Ranking interface

Our system also includes a web-based ranking interface, which displays the scores and corresponding ranking assigned to a set of annotated documents, as shown in Figure 6. For ease of visual distinction, we include colorization of rows based on configurable score thresholds. Ranking methods used for model scores and gold an-

Embeddings	Smoothing	Annotation			Ranking
		Pr	Rec	F-2	ρ
Static	No	59.0	94.7	84.4	0.862
	Yes	60.5	93.7	84.3	0.899
ELMo	No	60.2	94.1	84.4	0.771
	Yes	66.5	91.4	84.8	0.886
BERT	No	55.3	93.8	82.2	0.689
	Yes	62.3	90.8	84.3	0.844

Table 2: Annotation and ranking evaluation results on mobility documents, using three embedding sources. Results are given with and without Viterbi smoothing, using binarization threshold=0.5 and no collapsing of adjacent segments. Pr=precision, Rec=recall, ρ =Spearman’s ρ Pr/Rec/F2 are macro-averaged over folds, ρ is over all test predictions.

notations (when present) can be adjusted independently, and our post-processing methods (Section 3.2) can also be adjusted to affect ranking.

3.5 Qualitative analysis tools

We provide a set of three tools for performing qualitative analysis of annotation outcomes. The first measures lexicalization of each unique token in the dataset with respect to relevance score, by averaging the assigned relevance score (with or without smoothing) for each instance of each token. Tokens with a frequency below a configurable minimum threshold are excluded.

Our other tools analyze the aggregate relevance score patterns in an annotation set. For labeled data, as shown in Figure 2, we provide a visualization of precision, recall, and F-2 when varying the binarization threshold, including identifying the optimal threshold with respect to F-2. We also include a label-agnostic analysis of patterns in output relevance scores, illustrated in Figure 7, as a way to evaluate the confidence of the annotator. Both of these tools are provided at the level of an annotation set and individual documents.

3.6 Implementation details

Our automated annotation, post-processing, and document ranking algorithms are implemented in Python, using the NumPy and Tensorflow libraries. Our demonstration interface is implemented using the Flask library, with all backend logic handled separately in order to support modularity of the user interface.

4 Results on mobility

Table 2 shows the token-level annotation and document ranking results for our experiments on mo-

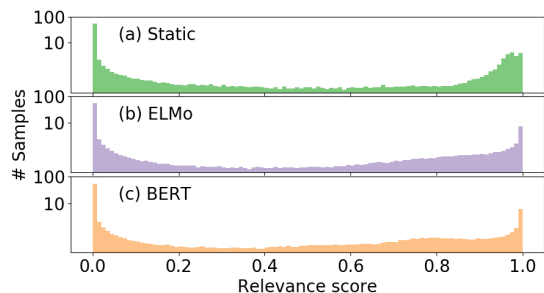


Figure 7: Distribution of token relevance scores on mobility data: (a) word2vec, (b) ELMo, and (c) BERT.

bility information. Static and contextualized embedding models performed equivalently well on token-level annotations; BERT embeddings actually underperformed static embeddings and ELMo on both precision and recall. Interestingly, static embeddings yielded the best ranking performance of $\rho = 0.862$, compared to 0.771 with ELMo and 0.689 with BERT. Viterbi smoothing makes a minimal difference in token-level tagging, but increases ranking performance considerably, particularly for contextualized models. It also produces a qualitative improvement by trimming out extraneous tokens at the start of several segments, as reflected by the improvements in precision.

The distribution of token scores from each model (Figure 7) shows that all three embedding models yielded a roughly bimodal distribution, with most scores in the ranges $[0, 0.2]$ or $[0.7, 1.0]$.

5 Discussion

Though our system is designed to address different needs from other NLP annotation tools, components such as annotation viewing are also addressed in other established systems. Our implementation decouples backend analysis from the front-end interface; in future work, we plan to add support for integrating our annotation and ranking systems into existing platforms such as brat. Our tool can also easily be extended to both multi-class and multilabel applications; for a detailed discussion, see Appendix C.

In terms of document ranking methods, it may be preferred to rank documents jointly instead of independently, in order to account for challenges such as duplication of information (common in clinical data; Taggart et al. (2015)) or subtopics. However, these decisions are highly task-specific, and are an important focus for designing ranking utility within specific domains.

6 Conclusions

We introduced HARE, a supervised system for highlighting relevant information and interactive exploration of model outcomes. We demonstrated its utility in experiments with clinical records annotated for narrative descriptions of mobility status. We also provided qualitative analytic tools for understanding the outcomes of different annotation models. In future work, we plan to extend these analytic tools to provide rationales for individual token-level decisions. Additionally, given the clear importance of contextual information in token-level annotations, the static transition probabilities used in our Viterbi smoothing technique are likely to degrade its effect on the output. Adding support for dynamic, contextualized estimations of transition probabilities will provide more fine-grained modeling of relevance, as well as more powerful options for post-processing.

Our system is available online at <https://github.com/OSU-slatelab/HARE/>. This research was supported by the Intramural Research Program of the National Institutes of Health and the US Social Security Administration.

References

- Emily Alsentzer, John Murphy, William Boag, Weihung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Clinical NLP Workshop*, pages 72–78. ACL.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *TACL*, 5:135–146.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. ACL.
- Ingrid E Fisher, Margaret R Garnsey, and Mark E Hughes. 2016. Natural Language Processing in Accounting, Auditing and Finance: A Synthesis of the Literature with a Roadmap for Future Research. *Intelligent Systems in Accounting, Finance and Management*, 23(3):157–214.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, pages 1–8.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, pages 1–12.
- Denis Newman-Griffis and Ayah Zirikly. 2018. Embedding Transfer for Low-Resource Medical Named Entity Recognition: A Case Study on Patient Mobility. In *BioNLP*, pages 1–11. ACL.
- Denis Newman-Griffis, Ayah Zirikly, Guy Divita, and Bart Desmet. 2019. Classifying the reported ability in clinical mobility descriptions. In *BioNLP*.
- Philip V Ogren. 2006. Knowtator: A Protégé plug-in for annotated corpus construction. In *NAACL-HLT*, pages 273–275, New York City, USA. ACL.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL-HLT*, pages 2227–2237, New Orleans, Louisiana. ACL.
- S Trent Rosenbloom, Joshua C Denny, Hua Xu, Nancy Lorenzi, William W Stead, and Kevin B Johnson. 2011. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *JAMIA*, 18(2):181–186.
- Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *JAMIA*, 17(5):507–513.
- Steven J Skube, Elizabeth A Lindemann, Elliot G Arsoniadis, Mari Akre, Elizabeth C Wick, and Genevieve B Melton. 2018. Characterizing Functional Health Status of Surgical Patients in Clinical Notes. In *AMIA Joint Summits*, pages 379–388. AMIA.
- Ergin Soysal, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, Hongfang Liu, and Hua Xu. 2018. CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines. *JAMIA*, 25(3):331–336.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *EACL*, pages 102–107. ACL.
- Jane Taggart, Siaw-Teng Liaw, and Hairong Yu. 2015. Structured data quality reports to improve EHR data quality. *Int J Med Info*, 84(12):1094–1098.

Thanh Thieu, Jonathan Camacho, and Pei-Shu Ho et al. 2017. Inductive identification of functional status information and establishing a gold standard corpus: A case study on the Mobility domain. In *BIBM*, pages 2300–2302. IEEE.

Yonghui Wu, Mike Schuster, and Zhifeng Chen et al. 2016. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint arXiv:1609.08144*.

A Hyperparameters

This section describes each of the settings evaluated for the various hyperparameters used in our experiments on mobility information. We first experimented with different pretrained embeddings used for each of our embedding model options; results are shown in Figure 8.

Static embedding model (Figure 8a) We evaluated three commonly used benchmark embedding sets: word2vec skipgram (Mikolov et al., 2013) using GoogleNews,⁶ FastText skipgram with subword information on WikiNews,⁷ and GloVe (Pennington et al., 2014) on 840 billion tokens of Common Crawl.⁸ Additionally, we experimented with two in-domain embedding sets, trained on 10 years of Physical Therapy and Occupational Therapy records from the NIH Clinical Center (referred to as “PT/OT”), using word2vec skipgram and FastText skipgram. word2vec GoogleNews embeddings produced the best dev F-2.

ELMo model (Figure 8b) We experimented with three pretrained ELMo models:⁹ the “Original” model trained on the 1 Billion Word Benchmark, the “Original (5.5B)” model trained with the same settings on Wikipedia and machine translation data, and a model trained on PubMed abstracts. The Original (5.5B) model produced the best dev F-2.

BERT model (Figure 8c) We experimented with three pretrained BERT models: BERT-Base,¹⁰ BioBERT (Lee et al., 2019) (v1.1) trained on 1 million PubMed abstracts,¹¹ and clinicalBERT (Alsentzer et al., 2019) trained on MIMIC data.¹² We use uncased versions of BERT-Base and clinicalBERT, as casing is not a reliable signal in clinical data; BioBERT is only available in a cased version. clinicalBERT produced the best dev F-2.

Once the best embedding models for each method were identified, we experimented with network and training hyperparameters, with re-

sults shown in Figure 9.

Irrelevant:relevant sampling ratio (Figure 9a) We experimented with the ratio of irrelevant to relevant samples drawn for each training epoch in 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3. A ratio of 0.75 gave the best dev F-2.

Positive fraction (Figure 9b) We varied the fraction of total dataset positive samples drawn for each training epoch from 10% to 100% at intervals of 10%. The best dev F-2 was produced by using all positive samples in each epoch.

Dropout rate (Figure 9c) We experimented with an input dropout rate from 0% to 90%, at intervals of 10%; the best results were produced with a 60% dropout rate.

Weighting scheme (Figure 9d) Given the imbalance of relevant to irrelevant samples in our dataset, we experimented with weighting relevant samples by a factor of 1 (equal weight), 2, 3, 4, and 5. A weighting of 2:1 produced the best dev F-2.

Hidden layer configuration (Figure 9e) We experimented with the configuration of our DNN model, using hidden layer size $\in \{10, 100, 300\}$ and number of layers from 1 to 3. The best dev F-2 results were achieved using 3 hidden layers of size 300.

B Ranking methods

A comparison of ranking methods used for model and gold scores is provided in Figure ???. We found that for our experiments, Segments+Tokens and SumScores correlated fairly well with one another, but Density, due to its normalization for document length, works best when used to rank both model and gold scores. SumScores provided the best overall ranking correlation; however, we use Segments+Tokens as the default setting for our system for its clear interpretation.

C Extending to multi-class/multilabel applications

Our experiments focused on binary relevance with respect to mobility information. However, our system can be fairly straightforwardly extended to both multi-label (i.e., multiple relevance criteria) and multi-class (e.g., NER) settings.

For multi-label settings, such as looking for evidence of limitations in either mobility or interpersonal interactions, the only requirement is having data that are annotated for each relevance cri-

⁶<http://google.com/archive/p/word2vec/>

⁷<https://fasttext.cc/docs/en/>

⁸<http://nlp.stanford.edu/projects/glove/>

⁹All downloaded from <https://allennlp.org/elmo>

¹⁰<https://github.com/google-research/bert>

¹¹<https://github.com/naver/biobert-pretrained>

¹²<https://github.com/EmilyAlsentzer/clinicalBERT>

terion. These can be the same data with multiple annotations, or different datasets; in either case, binary relevance annotators can be trained independently for each specific relevance criterion. Our post-processing components such as Viterbi smoothing can then be applied independently to each set of relevance annotations as desired. The primary extension required would be to the visualization interface, to support display of multiple (potentially overlapping) annotations. Alternatively, our modular handling of relevance annotations could be redirected to another visualization interface with existing support for multiple annotations, such as brat.

Extending to multi-class settings would require fairly minimal updates to both the interface and our relevance annotation model. Our model is trained using two-class cross (relevant and irrelevant) cross-entropy; this could easily be extended to n -ary cross entropy for any desired number of classes, and trained with token-level data annotated with the appropriate classes. In terms of visualization and analysis, the two modifications required would be adding differentiating displays for the different classes annotated (e.g., different colors), and updating the displayed evaluation statistics to micro/macro evaluations over the multiple classes. Qualitative analysis features such as relevance score distribution and lexicalization are already dependent only on the scores assigned to the “relevant” class, and could be presented for each class independently.

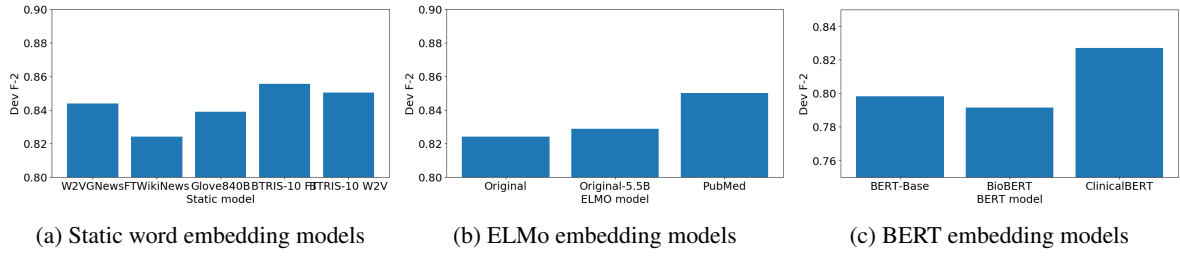


Figure 8: Embedding model selection results, by F-2 on cross validation development set. Default settings for other hyperparameters were: relevant:irrelevant ratio of 1:1, sampling 50% of positive samples per epoch, dropout of 0.5, equal class weights, and DNN configuration one 100-dimensional hidden layer.

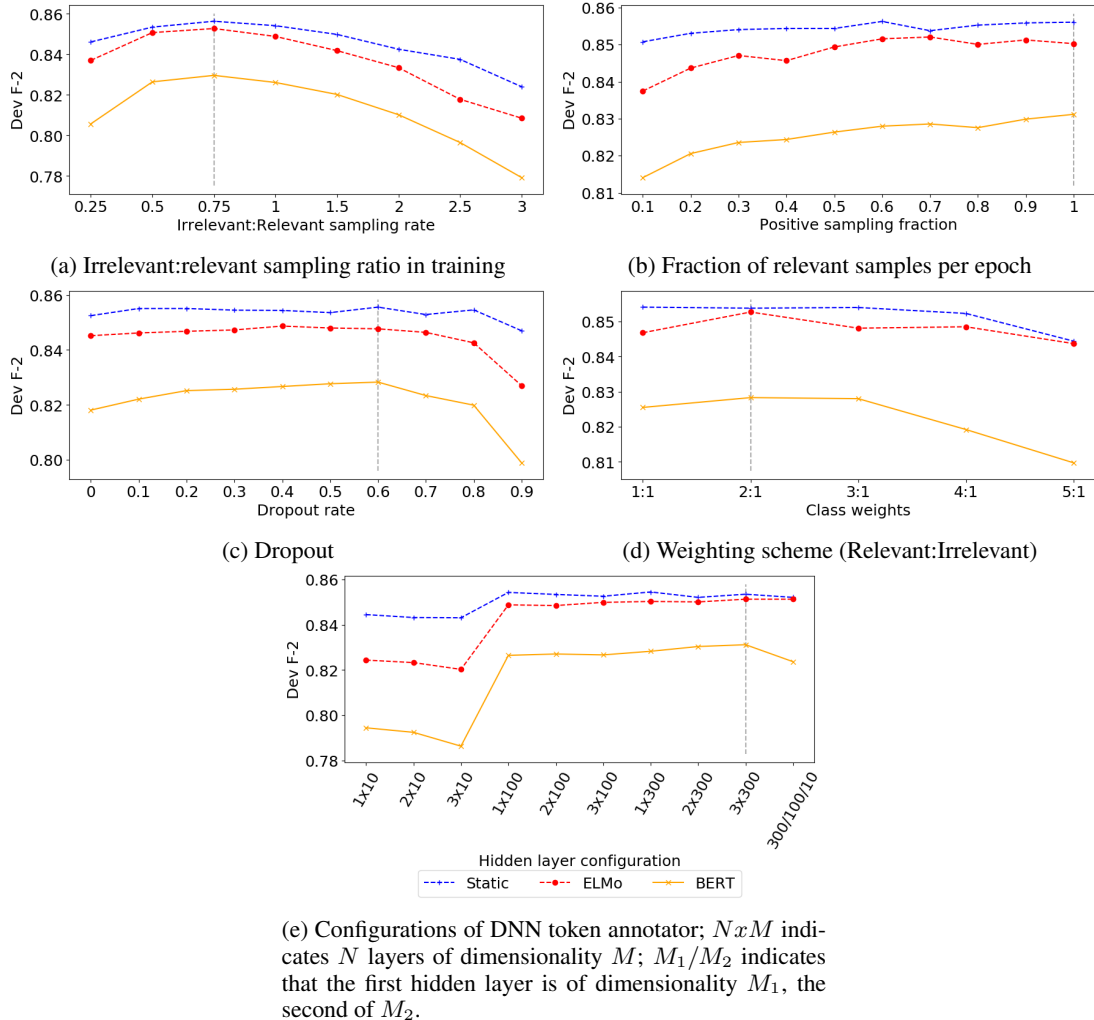


Figure 9: Hyperparameter tuning results, measuring F-2 on development set in cross-validation experiments. For each embedding method, the best model (shown in Figure 8) was used. All other hyperparameter setting defaults were as described in Figure 8. The best-performing setting for each hyperparameter, determined by the mean Dev F-2 across all three embedding methods, is indicated with a vertical dashed line.

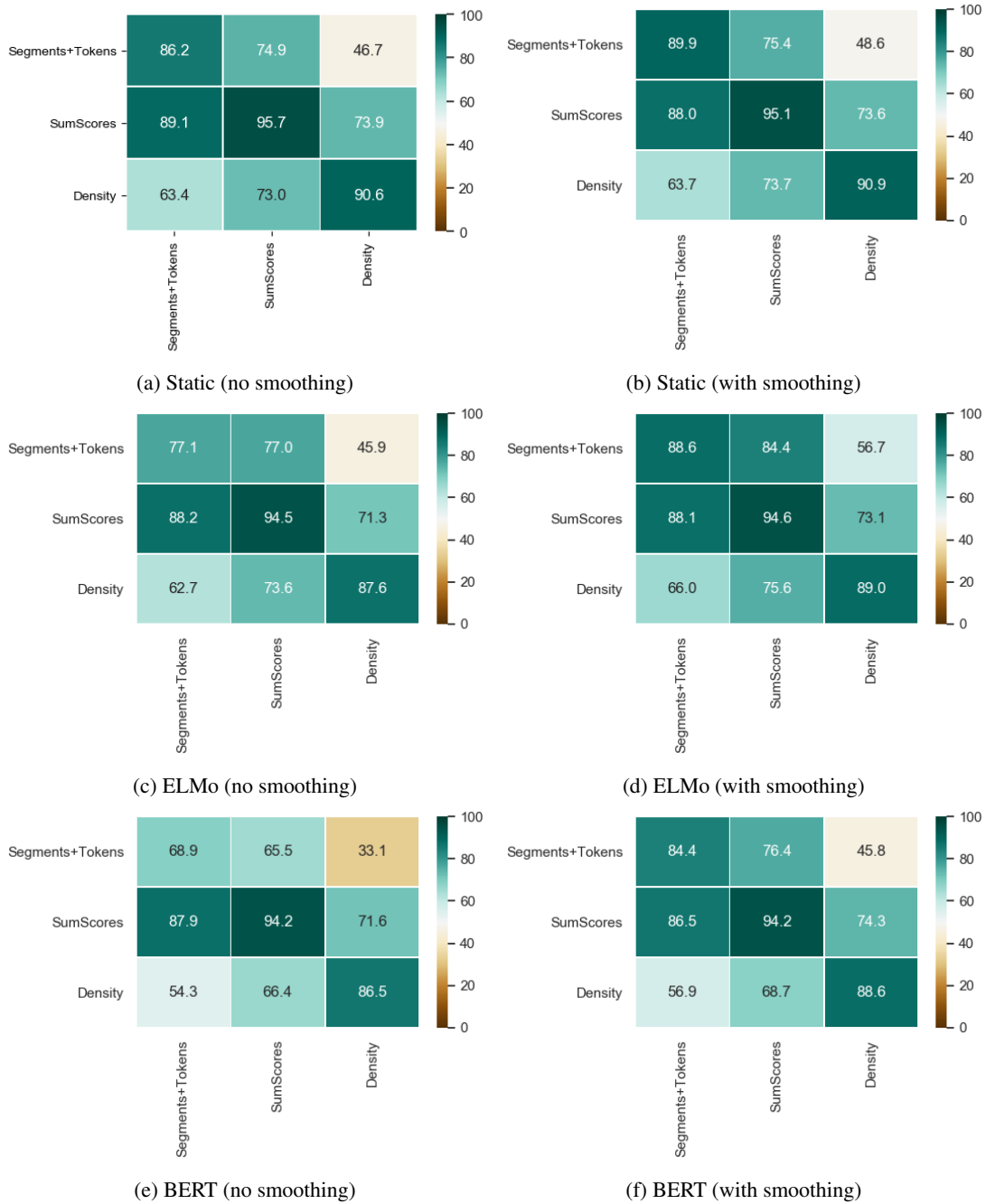


Figure 10: Comparison of ranking methods used for model scores and gold scores. Scores given are Spearman's rank correlation coefficient (ρ). System outputs using all three embedding methods are compared, both with and without Viterbi smoothing.