Content-based Models of Quotation

Ansel MacLaughlin
Khoury College of Computer Science
Northeastern University
Boston, MA
ansel@ccs.neu.edu

David A. Smith
Khoury College of Computer Science
Northeastern University
Boston, MA
dasmith@ccs.neu.edu

Abstract

We explore the task of quotability identification, in which, given a document, we aim to identify which of its passages are the most quotable, i.e. the most likely to be directly quoted by later derived documents. We approach quotability identification as a passage ranking problem and evaluate how well both feature-based and BERT-based (Devlin et al., 2019) models rank the passages in a given document by their predicted quotability. We explore this problem through evaluations on five datasets that span multiple languages (English, Latin) and genres of literature (e.g. poetry, plays, novels) and whose corresponding derived documents are of multiple types (news, journal articles). Our experiments confirm the relatively strong performance of BERT-based models on this task, with the best model, a RoBERTA sequential sentence tagger, achieving an average ρ of 0.35 and NDCG@1, 5, 50 of 0.26, 0.31 and 0.40, respectively, across all five datasets.

1 Introduction

Unlike in scientific writing, where authors use terse citations due to space constraints, direct quotation of source material is an essential part of writing in many fields. Journalists, humanities scholars, and students, for instance, often quote from a range of source documents, such as interviews, speeches, and books. Quotes can be used to substantiate a claim, lend authority to an argument, or offer a viewpoint to reflect on or argue against, among many others. Modeling the process of quotation selection is thus an important step in modeling how authors compose entire new documents.

In this paper, we explore the problem of quotability identification, identifying which passages in a source work (e.g. *Hamlet*) are likely to be quoted by later, derived works (e.g. humanities journal articles). Prior research has attempted to identify the specific factors that influence a passage or document's quotability. Most work, therefore, has focused on manual feature engineering and development of careful analysis frameworks to test which features have statistically significant relationships to quote counts (Tan et al., 2018; Danescu-Niculescu-Mizil et al., 2012). We, instead, reframe quotability identification as a practical passage ranking task and evaluate how well models can rank the passages in a given document by their predicted quotability. We benchmark and analyze the performances of multiple models, exploring both BERT-based passage ranking models and feature-based models equipped with "quotability" features identified in prior work.

We collect five large-scale datasets to study this problem. Each dataset consists of sets of source documents, derived documents, and alignments between them – the direct quotes. The proposed datasets are diverse, allowing us to model quotability dynamics across multiple source and derived document genres and languages.

Uses for our proposed passage ranking task include: 1) to help users discover quotable source content for use in an essay or article (Tan et al., 2016; MacLaughlin et al., 2021); 2) to select striking quotes from new books to help readers get a sense of them; 3) for use in extractive summaries, search result snippets, and other compressed versions of a text – since quotability implies the ability of a passage to have meaning outside its original source context, quotability scores could be incorporated with traditional measures of informativeness and non-redundancy to determine which passages should be excerpted.

The primary contributions of this paper include:

- We present five large-scale quotability identification datasets (§4) which span multiple

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2296–2314
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

source genres (novels, poems, plays, scripture, etc.) and languages (English, Latin).

- We compare the performance of multiple models (§6) on each of the five datasets: 1) feature-based models with bag-of-words and "quotability" features drawn from prior work (Bendersky and Smith, 2012) and 2) state-of-the-art BERT-based models for passage ranking and sequential sentence tagging. The best performing model, a RoBERTA-based sequential sentence tagger, achieves an average ρ of 0.35 and NDCG@1, 5, 50 of 0.26, 0.31 and 0.40, respectively, across all five datasets.
- We focus on a single source text, the King James Bible, to conduct a thorough analysis (§8.2). We analyze similarities and differences in both quoting patterns and modeling performance across two sets of aligned quoting labels – one from a large collection of 18th-20th century American Newspapers (11M pages), the other from the set of all journal articles in JSTOR (12M articles).

2 Related Work

There has been substantial previous research in identifying and analyzing what makes specific source content popular, i.e. how many times a given source is quoted, cited, retweeted, etc. The source documents analyzed in prior work span a wide range of domains, from political speeches and debates (Tan et al., 2018; Niculae et al., 2015) to books (Bendersky and Smith, 2012), movie scripts (Danescu-Niculescu-Mizil et al., 2012), scientific articles (Guerini et al., 2012; Yogatama et al., 2011), tweets (Hong et al., 2011; Tan et al., 2014), and news articles (Bandari et al., 2012). The aligned derived documents also span multiple domains, from social media (Booten and Hearst, 2016; Danescu-Niculescu-Mizil et al., 2012; Bendersky and Smith, 2012; Bandari et al., 2012; Hong et al., 2011; Tan et al., 2014) to news articles (Tan et al., 2018; Niculae et al., 2015), and scientific papers (Guerini et al., 2012; Yogatama et al., 2011).

Prior work has focused on both 1) predicting the popularity of an entire source document, e.g. a scientific article's citation count (Yogatama et al., 2011) and 2) similar to our work, identifying which specific passages in a given source work will receive the most attention (Tan et al., 2018; Danescu-Niculescu-Mizil et al., 2012; Bendersky and Smith, 2012). The most similar work to ours, Bendersky and Smith (2012), also explores applications of quotability identification models to literary works (Project Gutenberg). However, due to their lack of supervised data, they instead focus on the problem of identifying what sorts of passages are *likely to be* quoted, rather than modeling what sorts of phrases are *actually* quoted in derived works.

The contributions of prior work have been, primarily, feature engineering (e.g. number of personal pronouns, use of negative/positive words) and design of testing frameworks to determine which features have a statistically significant relationship to quotablity/popularity in a *single language* and in a *specific domain of interest* (e.g. English language movie quotes and their popularity on IMDB: Danescu-Niculescu-Mizil et al. (2012)). Unlike prior work, our main contributions are not feature engineering, but a re-framing of the task as a practical document-level ranking task and an analysis of several models through extensive experiments with multiple datasets spanning various source and derived document genres and languages.

3 Problem Formulation

We formulate quotability identification as passage ranking, identifying which passages in a source document are likely to be quoted in related derived works. We measure quotability directly by counting how many times a passage is quoted across a collection of derived texts. Concretely, given a source document and a set of derived texts, we –

1. Use fuzzy text alignment methods from text reuse detection, e.g. Smith et al. (2015), to identify alignments (quotes) between subsequences in the source and derived texts.
2. Split the source text into passages (e.g. prose: by sentence, poetry: by verse, plays: by line).
3. Map the starts and ends of the (source, derived) alignments, i.e. quotes, to specific source passages.
4. Label each source passage with the number of alignments that overlap it.
5. Learn to rank the passages by their quote count labels.
6. Measure how well a model can rank the passages in a source document w.r.t. each other.

4 Datasets

We have collected five datasets across two languages (English, Latin) and multiple genres (novels, poems, plays, scripture, etc.) to study this problem. Each dataset consists of source texts (e.g. Shakespeare's plays, books of the Bible) and a set of alignments to a corpus of derived documents (e.g. humanities journal articles, news articles). Each alignment is an instance where a derived work quotes a passage(s) in a specific source work, e.g. an author quotes from *Hamlet*, "To be, or not to be: that is the question," in a humanities journal article.

4.1 Source Document Datasets

We use four publicly available[1] source document datasets:

- **King James Bible (KJB)**: the standard English Bible from the mid-17th to the early 20th century, consisting of 66 books (Old and New Testaments, no Apocrypha), with an average of 471 verses per book (median 217).
- **Shakespeare (SHAK)**: Shakespeare's 38 plays, with an average of 3,284 lines per play (median 3,246).
- **American & British Literature (ABL)**: a collection of 70 American and British great works from the 17th-20th centuries, containing books (e.g. *Emma*), poetry (e.g. "Eve of St. Agnes"), essays (e.g. "On The Duty of Civil Disobedience"), speeches (e.g. "I Have a Dream"), and legal documents (e.g. US Constitution), with an average of 6,165 passages per work (median 5,457).
- **Latin Texts (LAT)**: a collection of 329 works of prose and poetry from the Perseus Digital Library (Crane, 2001), with an average of 1,832 sentences per document (median 853).

As seen in Table 1, there is substantial variation across the four datasets, both in terms of total numbers of documents and passages in each dataset and in median numbers of passages per document and tokens per passage. For example, LAT contains a relatively large number of documents and passages, but each document is relatively short (median 853 passages/doc). ABL, on the other hand, contains roughly 5x fewer documents, but each document contains roughly 6x more passages (median 5,457). Passages in KJB are the longest (median 27 tokens), but documents in KJB contain the smallest number of passages (median 217). See Appendix A for lists of the source documents in each dataset.

4.2 Derived Documents

We work with three collections of derived documents which discuss and quote from the above source documents (see Table 2 for more details):

- **Chronicling America (CA)**: A publicly available collection of roughly 11 million historic (1789-1963) newspaper pages from the Library of Congress's Chronicling America collection (Library of Congress, 2005).
- **JSTOR: Early Journal Content (EJC)**: A publicly available subset of the entire JSTOR collection, containing approximately 644k articles published prior to 1923 in the United States and prior to 1870 elsewhere.
- **JSTOR: All (JA)**: The entire JSTOR journal collection, consisting of over 12 million academic journal articles (not publicly available).

4.3 Source-Derived Alignments

We use three different sets of alignments between our source and derived document collections to generate labels for our datasets[2]:

- **KJB - CA**: alignments from America's Public Bible (Mullen, 2016). Using text reuse detection methods, Mullen (2016) identified quotations of the Bible or verbal allusions to specific biblical verses in newspapers from the Chronicling America collection. There are a total of 866,127 quotes from 383,387 unique pages across 1,706 different newspapers.
- **JSTOR Understanding Series**: alignments from the JSTOR Understanding Series (JSTOR Labs, 2019). The JSTOR Labs team created a database of all quotations within JSTOR, then, using text reuse detection methods, aligned those quotations to passages in a number of great works, including the King James Bible, Shakespeare, and American and British Literature datasets. JSTOR has provided us with the set of all alignments. There are a total of 65,093 quotes from 30,876 derived documents aligned to the Bible, 131,712 quotes from 24,060 derived documents aligned to Shakespeare's plays, and 130,582 quotes from

[1] KJB: Wikitext, ABL: Wikitext & Project Gutenberg, SHAK: Folger Shakespeare Library, LAT: Perseus

[2] The KJB-CA alignments (Mullen, 2016), the CA collection (Library of Congress, 2005) and the EJC are available for download. We also release processed, labeled versions of our KJB-CA and LAT-EJC datasets for download here. We are unable to release any JA data.

Source	Lang	Genre	Passage Type	# Docs	# Passages	Mdn # Passages / Doc	Mdn # Toks / Passage
KJB	en	Scripture	Verse	66	31,102	217	27
SHAK	en	Play	Line	38	124,809	3,246	9
ABL	en	Various	Verse, Sentence	70	431,580	5,457	19
LAT	la	Various	Sentence	329	602,676	853	13

Table 1: Summary statistics for the four source document datasets – King James Bible (KJB), Shakespeare's plays (SHAK), and collections of great works of literature from America and Britain (ABL) and the ancient Roman world (LAT).

Derived	Doc Type	# Documents
CA	newspaper pages	≈ 11,000,000
EJC	journal article	138,636
JA	journal article	≈ 12,000,000

Table 2: Summary statistics for the three derived document datasets – Chronicling America (CA), JSTOR: Early Journal Content (EJC) and JSTOR: All (JA).

28,986 derived documents aligned to the collection of American and British literature.

- **LAT - EJC**: We use the passim text alignment software (Smith et al., 2015) to detect quotes of the Latin texts in the JSTOR EJC using the Smith–Waterman alignment algorithm. This yielded a total of 124,679 aligned quotes from 26,619 derived journal articles.

As noted above, given a source document, a collection of derived documents, and set of alignments between the source and derived texts, we split the source into passages (sentences/lines/verses/etc) then count the number of times each passage is aligned to (in part or wholly) a distinct portion of a derived text. We then use these quote counts to label each passage in the source. See Appendix B for discussion of the implementation and accuracy of the different alignment-detection models.

As seen in Table 3, there is substantial variation across datasets with respect to the total number of aligned quotes and the proportion of source passages that are quoted. On one end of the spectrum, since there are many alignments between KJB and CA and KJB is a relatively small source (in terms of number of passages), 84% of source passages in the KJB-CA alignment dataset are quoted at least once, the median passage is quoted five times, and the median document contains a maximal passage quoted 480 times. On the other end of the spectrum, the ABL and LAT source datasets both contain a large number of passages, and there are relatively few alignments, leading to significantly sparser ABL-JA and LAT-EJC alignment datasets. Only 18% (ABL) and 25% (LAT) of passages are quoted at least once, and the median documents contain maximal passages quoted 43 (ABL) and 7 (LAT) times. In the middle are the KJB-JA and SHAK-JA datasets, where slightly over half of the passages are quoted at least once, the passages in the 75th percentile are quoted three times, and the median documents contain maximal passages quoted 30 (KJB-JA) and 53 (SHAK-JA) times.

5 Linguistic Attributes of Quotations

As a first step at modeling the quotability identification problem, we draw on quotability features from prior work (Bendersky and Smith, 2012; Danescu-Niculescu-Mizil et al., 2012; Tan et al., 2018) and attempt to identify which linguistic attributes influence passage quotability. We explore both lexical (e.g. passage begins with a stop word) and part of speech features (e.g. contains past tense verb).

For each dataset, we featurize each of the corresponding passages, then compare its highly-quoted (top 20%) and minimally-quoted (bottom 20%) passages. We use Welch's t-test with Bonferroni correction to test whether the scores for each feature are significantly different between the two groups.

Table 4 discusses each feature and the associated statistic and significance for each dataset. Most features' relationship with quotability varies across the five datasets. For instance, highly-quoted passages are *shorter* in KJB-CA, but *longer* in other datasets. We suspect this reflects a difference between a popular audience (CA) who might prefer short Bible verses with a succinct message vs. academia (JSTOR), where writers are focused on careful passage analysis and less space-constrained. We also observe differences between the two KJB datasets and the other datasets. In the KJB, the relationship between quotability and presence of dialogue words (says, etc.) is positive, but negative in the others. This difference may be because many important Bible verses report direct speech by Jesus or God, whereas the other datasets, e.g. ABL, contain many uninteresting dialogue passages that serve to move the story along (e.g. "'No,' said the boy").

Source	Derived	# Aligned Derived Quotes	# Aligned Derived Docs	# Quotes / Source Passage Q_1	 Q_2	 Q_3	 Max	% Passages w/ $\geq$ 1 Quote	Mdn Max # Quotes / Doc
KJB	CA	866,127	383,387	1	5	19	4,949	84%	480
KJB	JA	65,093	30,876	0	1	3	303	58%	30
SHAK	JA	131,712	24,060	0	1	3	905	58%	53
ABL	JA	130,582	28,986	0	0	0	1,536	18%	43
LAT	EJC	124,679	26,619	0	0	0	62	25%	7

Table 3: Summary statistics for the five sets of source-derived alignments. Each set of alignments is between one source dataset (KJB, SHAK, ABL, LAT) and one derived document dataset (CA, JA, EJC). Each source passage is labeled by the total number of times it is quoted in the corresponding derived documents. # Quotes / Source Passage measures of how many quotes each source passage receives - we display the first, second and third quartiles and the max for passages across the entire source dataset. % Passages w/ $\geq$ 1 Quote measures what percentage of passages in the source dataset have at least 1 aligned quote. To calculate Mdn Max # Quotes / Doc, we find the most quoted passage in each document in the source dataset, then take the median over those quote counts.

Feature Set		KJB-CA	KJB-JA	SHAK-JA	ABL-JA	LAT-EJC
Length: Number of tokens in passage. (Bendersky and Smith, 2012)	# Words	↓↓↓↓	↑↑↑↑	↑↑↑↑	↑↑↑↑	↑↑↑↑
Capitalized: Proportion of words capitalized (Bendersky and Smith, 2012).	% Capital	↓↓↓↓	↓↓↓↓	↓↓↓↓	↓↓↓↓	↑↑↑↑
Stop Words: 1) Proportion of words in passage that are stop words, 2) Binary feature if passage begins with a stop word (Bendersky and Smith, 2012).	% Stop Begin-Stop	– –	↑↑↑↑ –	↑↑↑↑ –	↑↑↑↑ ↑↑	↓↓↓↓ ↓↓↓↓
Generality: 1) Proportion of words in passage that are indefinite articles 2) Binary feature if passage contains an abstract noun (Bendersky and Smith, 2012; Danescu-Niculescu-Mizil et al., 2012; Tan et al., 2018).	% Indefinite Abstract	↑↑↑↑ ↑↑↑↑	↑↑↑↑ ↑↑↑↑	↑↑↑↑ ↑↑↑↑	↑↑↑↑ ↑↑↑↑	× ×
Universal Quantifiers: Binary feature if passage contains a universal quantifiers (20 quantifiers, e.g., all, whole, nobody). (Bendersky and Smith, 2012).	Universal	↑↑↑↑	↑↑↑↑	↑↑↑↑	↑↑↑↑	×
Pronouns: Proportion of words in passage that are first, second, or third person pronouns (Tan et al., 2018).	% 1st % 2nd % 3rd	↑↑↑↑ ↑↑↑↑ –	↑↑↑↑ ↑↑↑↑ ↓↓↓↓	– ↓↓↓↓ ↓↓↓↓	↓↓↓↓ ↓↓↓↓ ↓↓↓↓	↑↑↑↑ ↑↑↑↑ ×
Language Model: We compute the ratio between the log-likelihood of the passage under a LM trained on a collection of popular quotes and one trained on a background corpus (Bendersky and Smith, 2012, eq. 1). The more likely a passage is under the quotable LM relative to the background LM, the higher the ratio will be (Danescu-Niculescu-Mizil et al., 2012; Tan et al., 2018).	LLR	↑↑↑↑	↑↑↑↑	↑↑↑↑	↑↑↑↑	×
Dialogue Words: Binary feature if passage contains a dialogue word (e.g. says, dicit). (Bendersky and Smith, 2012)	Dialogue	↑↑↑↑	↑↑↑↑	↓	↓↓↓↓	↓↓↓↓
Emphasis: 1) Binary feature if passage contains a comparative adjective or adverb form, 2) Binary feature if passage contains a superlative adjective or adverb form (Bendersky and Smith, 2012; Tan et al., 2018).	Comparative Superlative	↑↑↑↑ ↑	↑↑↑↑ –	↑↑↑↑ ↑↑↑↑	↑↑↑↑ ↑↑↑↑	↑↑↑↑ ↑↑↑↑
Verb Tenses: Proportion of words in passage that are past or present tense verbs (Bendersky and Smith, 2012; Danescu-Niculescu-Mizil et al., 2012).	Past Present	– ↑↑↑↑	↓↓↓↓ ↑↑↑↑	↑↑↑↑ ↓↓↓↓	↓↓↓↓ –	↑↑↑↑ ↓↓↓↓

Table 4: Results from examination of linguistic attributes of quotations. Upward arrows indicate that frequently quoted passages (top 20% of source passages by # of quotes) have larger scores in that feature, while downward arrows indicate that passages with very few or no quotes (bottom 20% of source passage by # of quotes) have larger scores in the feature (↑↑↑↑: $p < 0.0001$, ↑↑↑: $p < 0.001$, ↑↑: $p < 0.01$, ↑: $p < 0.05$). p refers to the p-value after the Bonferroni correction. × indicates that the feature set is not available for the corresponding dataset. A '–' symbol indicates that there is no significant relationship.

While there are no features whose relationship with quotability is negative across all datasets, there are a few that are consistently significantly positive: **1) general language:** highly-quoted passages contain more general language which can be more easily adapted to a new contexts, such as indefinite articles and abstract concepts (e.g. adventure, charity); **2) universal quanitifiers:** similar to maxims, proverbs and other short, pithy statements, quotable phrases often contain universal quantifiers (e.g. always, never); **3) comparative adjectives:** highly-quoted passages use more comparative adjectives to compare a noun to something else; **4) language model**: quotable phrases have higher likelihoods under our quotable language model than a background language model trained on the corresponding source corpus.

6 Models

Next, we examine the effectiveness of two classes of models on our passage ranking task: feature-based models with "quotability" features and neural models for sentence classification. We benchmark both pointwise regression models for count data and listwise and pairwise ranking models.

6.1 Feature-based

We evaluate three feature-based models: **poisson regression**, **SVM^{rank}** (Joachims, 2002) and **lambdaMART** (Burges, 2010). Poisson regression (PR) is a generalized linear model for count data, SVM^{rank} is a commonly used pairwise feature-based ranker, and lambdaMART (λMART) is a state-of-the-art feature-based listwise ranker. Each feature-based model uses the same set of features: **bag-of-word features**, **"quotability" features** (Bendersky and Smith, 2012; Danescu-Niculescu-Mizil et al., 2012; Tan et al., 2018, Including the features discussed in §5), and **positional features** (e.g. relative position in book). See Appendix E for a list of all features.

6.2 BERT-based

BERT-based models (Devlin et al., 2019) have recently achieved state-of-the-art performance on multiple sentence classification tasks, including single sentence classification, e.g. sentiment analysis, and classification of a sequence of sentences into their corresponding categories, e.g. extractive summarization (Liu and Lapata, 2019) and scientific abstract sentence classification (Cohan et al., 2019). We work with two BERT-based models, RoBERTA (Liu et al., 2019) and XLM-RoBERTA (XLM-R) (Conneau et al., 2019). We fine-tune RoBERTA on our English datasets and XLM-R on the Latin dataset. We benchmark both models as both single passage and sequential passage predictors:

- **RoBERTA**$_{single}$ **/ XLM-R**$_{single}$: we fine-tune RoBERTA and XLM-R on individual passages. Thus, each training example contains a single passage and its quote count label. We follow the standard fine-tuning setup, using the final hidden state of the [`CLS`] token as the aggregate representation for a passage and feeding it into an output layer for prediction.

- **RoBERTA**$_{seq}$ **/ XLM-R**$_{seq}$: in order to model a passage's context in the broader document, we also fine-tune RoBERTA and XLM-R as passage-level sequence taggers (Cohan et al., 2019). Due to the models' 512 WordPiece length limit, we break up each document into 512 WordPiece segments and feed those into the models independently. Thus, each training example contains some number n of consecutive passages from the same work (up to 512 WordPieces total) and n quote count labels, one for each passage. Following Cohan et al. (2019), we insert a [`SEP`] token between each of the n passages in each example. We use the final hidden state of each [`SEP`] token as the aggregate representation for each passage and feed it into a multi-layer feedforward network to make a prediction.

Since we are modeling quote counts for each passage, we train all $single$ and seq models with poisson negative log likelihood loss.

Finally, since our task is a ranking task, we also benchmark a BERT-based pairwise ranker, as pairwise neural rankers have shown strong performance on other ranking tasks, such as ad-hoc retrieval (Xiong et al., 2017; Dai et al., 2018):

- **RoBERTA**$_{pair}$ **/ XLM-R**$_{pair}$: Each training example consists of two passages sampled from the same work and a single label for the pair based on which passage is quoted more. Each passage is fed into RoBERTA or XLM-R separately and follows the standard fine-tuning setup as described for RoBERTA$_{single}$.

We train RoBERTA$_{pair}$ and XLM-R$_{pair}$ with hinge loss: $\mathcal{L}(s^+, s^-; \Theta) = max(0, 1 - f(s^+) + f(s^-))$, where passages s^+ and s^- are from the same document, s^+ is quoted more than s^-, and $f(s)$ is the output of running passage s through RoBERTA/XLM-R and the final output layer.

For all neural models, we also add special tokens to each passage to act as positional indicators so the model has a better sense of which part of the document it is reading (Alberti et al., 2019). These special tokens vary by dataset, but are generally added to the start or end of a passage and have forms such as [`Starts-Paragraph`], [`Ends-Act`], or [`Book@N`] where `N` is the decile in the book in which the passage occurs. See Appendix F for the full list of positional tokens used in each dataset.

7 Experimental Settings

We evaluate models using five-fold cross validation. We train and evaluate models on each dataset separately and report means over the five folds for each. We split datasets into folds at the document level (e.g. train: *Frankenstein*, etc., val: *Emma*, etc., test: *Paradise Lost*, etc.). Then, for a given validation or test set, we evaluate performance on each document separately, then average over documents.

For the feature-based models, we select hyperparameters by performing nested five-fold cross validation on each fold's training set, again splitting by document. Due to computational restrictions, for the neural models we select hyperparameters by using 20% of the documents in the fold's training set as a validation set. See Appendix H for a list of all hyperparameters.

7.1 Evaluation Metrics

Since our task is a passage ranking task, we evaluate models using two common ranking metrics: average NDCG@k (Croft et al., 2009) and spearman's ρ. NDCG@k is defined as $\frac{DCG_k}{IDCG_k}$, where $DCG_k = quoted_1 + \sum_{i=2}^{k} \frac{quoted_i}{log_2(i)}$ and $quoted_i$ is the number of times the passage ranked in the ith position by the model has been quoted across all corresponding derived documents, and $IDCG_k =$ the ideal DCG_k, the maximum DCG_k computed by ranking passages by their true quote counts. We evaluate NDCG@k at six ranks, $k \in [1, 3, 5, 10, 25, 50]$. We calculate both NDCG and ρ at the document level, i.e. ranking a passage in a document versus all other passages in that document, then average across documents.

8 Results & Analysis

Tables 5 and 6 display the results of all models across all five datasets[3]. On the whole, we find that modeling a passage's context in its broader document is important, with the BERT-based sequential sentence models RoBERTA$_{seq}$ and XLM-R$_{seq}$ performing the best. They achieve the highest average NDCG score across all ranks k on three datasets (KJB-CA, SHAK-JA, LAT-EJC) and the highest ρ on all five datasets. The sequential sentence models achieve their highest *relative* performance on SHAK-JA, outperforming the second-best models by roughly 40% relative on NDCG and 19% on ρ.

Similar to results on other passage ranking tasks (Nogueira and Cho, 2019; Qiao et al., 2019), we find that the single passage BERT-based models (RoBERTA$_{single}$, XLM-R$_{single}$) provide strong baselines, outperforming the feature-based models on nearly all datasets and achieving the second-best overall performance. Furthermore, on KJB-JA, RoBERTA$_{single}$ achieves the best NDCG performance, outperforming RoBERTA$_{seq}$ by an average of 0.02 across each k. Investigating the KJB-JA results further, we find that Roberta$_{single}$ outperforms Roberta$_{seq}$ on 34 of the 64 KJB books. Of these 34, five books (Numbers, Revelation, Zechariah, Leviticus, and Peter-1) account for over a third of the total increase in NDCG over Roberta$_{seq}$. In four of the five books, Roberta$_{single}$ successfully ranks the most quoted passage at the top, while Roberta$_{seq}$ ranks passages with single- or near-single-digit labels. Notably, Roberta$_{seq}$ fails, while Roberta$_{single}$ succeeds, in properly ranking the Great Commandment from Leviticus 19:18 "... love thy neighbour as thyself" at the top.

On the other hand, the pairwise BERT-based models (RoBERTA$_{pair}$, XLM-R$_{pair}$) generally perform worse than the single passage and sequential passage BERT-based models. They struggle to identify the top, most quoted passages in each document (as measured by NDCG), but perform relatively better at ranking each document's entire list of passages w.r.t each other (ρ), though still worse than the sequential passage models.

Among the feature-based models, while SVM^{rank} achieves the highest ρ on all five datasets, no single model consistently outperforms the others in NDCG, with PR, SVM^{rank}, and λMART each achieving the highest scores on different datasets. However, only one feature-based model, λMART, ever outperforms the neural models, achieving the highest NDCG scores on ABL-JA. We hypothesize that λMART's strong performance on ABL-JA might be due, in part, to differences in the accuracy of our feature-extraction pipeline. Many of our features depend on accurate parsing (e.g. POS and verb tense counts). However, for our English datasets, the Stanza Universal Dependencies model (Qi et al., 2020) we use to process each sentence is trained on web-media data (UD English EWT). Thus, our English datasets (KJB, ABL, SHAK) are all out-of-domain. We hypothesize that Stanza is more accurate on ABL since it contains the most modern language similar to its training data. With these higher quality inputs, therefore, feature-based models can achieve higher performance, relative to the neural models. As one might note, our datasets are also out-of-domain for RoBERTa and XLM-R. However, as shown by Han and Eisenstein (2019), BERT-based models can adapt to new domains when provided with in-domain fine-tuning data. For our Latin data, although our LAT-EJC dataset is in-domain for the Latin Stanza model (UD Latin Perseus), the Stanza model might not perform as well since the UD training data is quite small.

8.1 Differences in Model Performance Across Datasets

As seen in Tables 5 and 6, performance of individual models varies substantially across the five datasets. Examining the average NDCG@k scores,

[3] There are only 64 total documents in KJB-JA since there are no aligned quotes in John-2 or Kings-2.

	KJB - CA (66 books)							KJB - JA (64 books)						
	Average NDCG@							Average NDCG@						
	1	3	5	10	25	50	ρ	1	3	5	10	25	50	ρ
PR	19.5	25.8	26.9	30.0	35.8	41.1	27.3	32.3	33.0	33.6	34.7	39.7	45.3	21.1
SVM^{rank}	12.7	17.0	19.7	23.1	29.9	35.4	33.4	21.6	23.0	24.7	28.8	34.3	41.1	23.9
λMART	18.4	18.1	20.2	22.7	28.6	34.8	23.4	13.3	18.6	21.4	25.2	31.7	38.1	20.4
RoBERTA$_{single}$	**24.8**	27.1	30.4	34.6	41.2	46.0	41.0	**34.3**	**35.3**	**35.9**	**38.5**	**44.3**	**49.8**	30.9
RoBERTA$_{pair}$	19.3	26.0	28.9	32.0	37.5	43.1	39.8	25.8	25.6	27.1	30.8	38.1	44.1	29.2
RoBERTA$_{seq}$	23.1	**31.4**	**34.7**	**37.2**	**43.1**	**47.7**	**42.0**	31.9	31.3	33.7	37.7	43.6	49.2	**34.1**

	ABL - JA (70 works)							SHAK - JA (38 plays)						
	Average NDCG@							Average NDCG@						
	1	3	5	10	25	50	ρ	1	3	5	10	25	50	ρ
PR	22.1	21.7	22.9	23.6	25.0	26.8	20.9	9.4	11.4	12.3	14.0	16.4	18.7	23.9
SVM^{rank}	17.2	20.6	21.7	22.9	25.2	27.3	25.7	11.5	12.8	13.0	14.4	16.5	19.2	35.7
λMART	**31.1**	**27.6**	**28.6**	**30.6**	**31.2**	**32.8**	24.3	9.3	10.4	10.8	12.4	14.7	16.9	29.8
RoBERTA$_{single}$	16.3	20.5	21.7	23.8	26.0	28.8	24.6	10.6	18.3	19.2	20.2	22.8	25.1	38.6
RoBERTA$_{pair}$	20.1	22.5	22.9	24.3	27.6	30.0	28.2	11.1	16.1	17.1	18.5	21.0	23.7	38.2
RoBERTA$_{seq}$	16.5	23.2	24.2	26.0	29.2	32.2	**31.5**	**24.8**	**23.6**	**25.0**	**27.6**	**30.2**	**33.5**	**46.7**

Table 5: 5-fold cross validation results on the King James Bible, with both Chronicling America (CA) and JSTOR All (JA) alignments, and on the American and British Literature (ABL) and Shakespeare (SHAK) datasets, both with alignments from JSTOR All (JA). We report NDCG across six positions (1, 3, 5, 10, 25, 50). Reported NDCG and ρ values are averaged across documents within each fold then averaged across folds.

a measure of how well a model identifies the top k most quoted passages in a given document, we find that models struggle most with the Shakespeare dataset, SHAK-JA, and achieve the best scores on the King James Bible with alignments from JSTOR, KJB-JA. On the other hand, examining the ρ scores, a measure of how well a model ranks *all* of the passages in a given document with respect to each other, we find that models struggled most on the Latin dataset, LAT-EJC, and achieve their highest scores on the Shakespeare dataset, SHAK-JA.

We hypothesize that these differences in modeling performance might be due, in part, to high-level, non-linguistic differences between the datasets, such as the average number of passages in each source work. First, examining NDCG scores, we find that scores are generally higher on datasets where the documents have relatively few passages (KJB, LAT) than on those where the documents contain many passages (SHAK, ABL). Since NDCG evaluates how well models identify the top k passages in a given document, this relative performance difference is understandable because shorter passage lists are likely easier to rank than very long ones (e.g. 155 verses in Ephesians vs. 9,426 sentences in Moby Dick). On the other hand, we do not find clear relationships between NDCG and either 1) the proportion of passages that are quoted at least once; or 2) the size of the dataset (total # passages). Models have relatively similar performances on 1) both sparsely quoted (ABL-JA) and highly quoted (SHAK-JA, KJB-CA) datasets; and 2) both large (LAT-EJC) and small (KJB-JA) datasets.

	LAT - EJC (329 works)						
	Average NDCG@						
	1	3	5	10	25	50	ρ
PR	28.0	27.8	27.6	28.2	29.5	31.0	12.1
SVM^{rank}	25.5	25.0	26.6	27.7	29.8	31.9	16.3
λMART	32.5	29.9	29.6	30.0	31.3	32.7	14.2
XLM-R$_{single}$	29.4	30.2	30.1	30.6	32.0	33.6	14.1
XLM-R$_{pair}$	25.1	26.2	27.0	27.9	29.5	31.5	16.5
XLM-R$_{seq}$	**35.8**	**35.6**	**36.3**	**35.8**	**37.0**	**38.7**	**19.4**

Table 6: 5-fold cross validation results on the collection of Latin texts (LAT) with alignments from the JSTOR Early Journal Collection (EJC). We report NDCG across six positions (1, 3, 5, 10, 25, 50). Reported NDCG and ρ values are averaged across documents within each fold then averaged across folds.

Finally, inspecting differences in ρ scores across datasets, we find that although scores are generally higher on datasets with a high proportion of quoted passages (SHAK-JA, KJB-CA), this trend does not always hold – models have approximately equal ρ on KJB-JA and ABL-JA, though 40% more of the passages in KJB-JA (58%) are quoted at least once compared to ABL-JA (18%).

8.2 King James Bible: Analysis

We conduct a thorough analysis on a single source text, the King James Bible. We select KJB since it is aligned to two separate derived document collections, Chronicling America (KJB-CA) and JSTOR: All (KJB-JA), allowing us to conduct comparative analysis across two different labelings of the same source document. We focus on differences in quoting attention and relative modeling performances.

Quoting Attention: We first examine differences in quoting attention aggregated at the book level. To identify which books of the Bible receive the most attention, we rank them by both their median and maximum number of quotes per passage. Table 7 lists the top 3 books by both metrics (mdn & max) for each dataset. There is only one book in common across both lists, the Gospel of Luke. The top 6 books in CA are all from the New Testament, while the top 6 in JA are split evenly between the Old and New Testaments, perhaps indicating a difference between a popular (CA) and scholarly (JA) audience. We calculate the overall similarity in book-level quoting attention between KJB-JA and KJB-CA by computing correlation (ρ) between the datasets' aggregate quote counts: ρ is 0.63 ranking by max passage and 0.62 by median.

Next, we compare differences in quoting attention over specific passages. We iterate over each book and compute ρ between the passage-level quote counts from each dataset. The average ρ across the 64 books[4] is 0.40. The books with the most similar and dissimilar quoting attention are Philippians and Nahum, respectively.

Modeling Performance: We focus on RoBERTA$_{seq}$ for our analysis since it achieves the best ρ on both datasets and the best NDCG on KJB-CA and second best on KJB-JA. We focus on relative performances at the book level. For each Bible book in each alignment dataset, we compute a composite model score by averaging RoBERTA$_{seq}$'s ρ and NDCG (averaged across all k) scores. We then compute the correlation (ρ) between this model score for each book and the book's 1) *length:* book length, in total # of passages 2) *proportion*: proportion of passages quoted at least once 3) *median:* median quote label 4) *max:* maximum quote label, and 5) *entropy*: entropy of the distribution of quotes over passages. On the whole, we find that RoBERTA$_{seq}$ performs better (i.e. higher model scores) on books with high median quote labels (*median*) and many quoted passages (*proportion*). Specifically, under KJB-JA labels, we find positive correlation between model score and *median* (0.42), *max* (0.42), and *proportion* (0.34) and no correlation with *length* or *entropy*. Under KJB-CA labels, we find weaker, positive correlation between model score and *median* (0.23) and *proportion* (0.21), no correlation with *max*, and negative correlation with *length* (-0.27) and *entropy* (-0.26).

	JA	**CA**
Max	John, Genesis, Luke	Matthew, Mark, Luke
Mdn	Song of Solomon, Revelation, Jonah	James, First John, Ephesians

Table 7: The top 3 most quoted books of the King James Bible as measured by **Max**: the maximum quoted passage in each book; **Mdn**: the median quote count in each book. We compare quote counts from alignments between the KJB and both JSTOR All (JA) and Chronicling America (CA).

Finally, we make a comparison between model scores across the two datasets, computing ρ between the two scores for each book. We find that RoBERTA$_{seq}$ performs relatively similarly on books across the two datasets, with moderate correlation of 0.45 between the two sets of scores.

9 Conclusion

We explore the task of quotability identification – identifying which passages in a source document are likely to be directly quoted by later derived documents. We cast quotability identification as a passage ranking problem, evaluating how well models can learn to rank the passages in a source document by their predicted quotability. We evaluate on five large-scale datasets spanning multiple source genres (e.g. poetry, novels, plays) and languages (English, Latin). We conduct experiments with feature- and BERT-based models, finding that although relative performances vary across datasets, on the whole, BERT-based models operating on strings of sequential passages perform best.

We have identified two potential directions of future research using the datasets described in this study. First, using the publication date information for the journal articles and newspapers, we could investigate temporal quoting trends, testing hypotheses such as the Matthew effect, and therefore the best predictor of a passage's quotability tomorrow is its popularity today. Finally, we could explore second-order effects, studying trends in what sorts of passages are often quoted together.

Acknowledgements

Ansel MacLaughlin was supported by a National Endowment for the Humanities Digital Humanities Advancement Grant (HAA-263837-19).

[4] We ignore John-2 and Kings-2 since they are not quoted in KJB-JA.

References

Chris Alberti, Kenton Lee, , and Michael Collins. 2019. A bert baseline for the natural questions. In *arXiv preprint arXiv:1901.08634*.

Roja Bandari, Sitaram Asur, and Bernardo A Huberman. 2012. The pulse of news in social media: Forecasting popularity. In *Proceedings of ICWSM*.

Michael Bendersky and David Smith. 2012. A dictionary of wisdom and wit: Learning to extract quotable phrases. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 69–77, Montréal, Canada. Association for Computational Linguistics.

Kyle Booten and Marti A. Hearst. 2016. Patterns of wisdom: Discourse-level style in multi-sentence quotations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1139–1144, San Diego, California. Association for Computational Linguistics.

Chris J.C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. Technical report, Microsoft Research.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, F. Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *ArXiv*, abs/1911.02116.

Gregory R. Crane. 2001. Perseus digital library. Tufts University. `http://www.perseus.tufts.edu`.

W. Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines - Information Retrieval in Practice*. Pearson.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. You had me at hello: How phrasing affects memorability. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Jeju Island, Korea. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Guerini, Alberto Pepe, and Bruno Lepri. 2012. Do linguistic style and readability of scientific abstracts affect their virality? In *Proceedings of ICWSM*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

Liangjie Hong, Ovidiu Dan, and Brian D Davison. 2011. Predicting popular messages in twitter. In *Proceedings of WWW*.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*.

JSTOR Labs. 2019. Jstor understanding series. `https://guides.jstor.org/understandingseries`.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Library of Congress. 2005. Chronicling america: Historic american newspapers site. `https://chroniclingamerica.loc.gov/`.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arxiv preprint: 1907.11692*.

Ansel MacLaughlin, Tao Chen, Burcu Karagol Ayan, and Dan Roth. 2021. Context-based quotation recommendation. *ICWSM*.

Lincoln Mullen. 2016. America's public bible: Biblical quotations in u.s. newspapers, website, code, and datasets. http://americaspublicbible.org.

Vlad Niculae, Caroline Suen, Justine Zhang, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2015. Quotus: The structure of political media coverage as revealed by quoting patterns. In *WWW*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *ArXiv*, abs/1904.07531.

D. A. Smith, David R. Cordell, and David A. Ryan Abby Mullen. 2015. Computational methods for uncovering reprinted texts in antebellum newspapers. *American Literary History*, 27:E1 – E15. https://github.com/dasmiq/passim.

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 175–185, Baltimore, Maryland. Association for Computational Linguistics.

Chenhao Tan, Hao Peng, and Noah A. Smith. 2018. "You are no Jack Kennedy": On media selection of highlights from presidential debates. In *WWW*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *CIKM*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russel Power. 2017. End-to-end neural adhoc ranking with kernel pooling. In *SIGIR*.

Shaobin Xu, David Smith, Abigail Mullen, and Ryan Cordell. 2014. Detecting and evaluating local text reuse in social networks. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 50–57, Baltimore, Maryland. Association for Computational Linguistics.

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 594–604, Edinburgh, Scotland, UK. Association for Computational Linguistics.

A Source Datasets

Below are lists of all source documents in each of the four source datasets. For the sake of space, the names of all 329 works in the LAT datset are detailed here (see latin-works-metadata.jsonl).

The books in KJB are separated into passages by verse. The plays in SHAK are separated into passages by line. The documents in ABL and LAT are separated into passages by using Stanza's (Qi et al., 2020) sentence tokenizer. For the poems in ABL, we use the standard line breaks rather than sentences.

- **KJB, the 66 books of the Old and New Testaments**: acts, amos, chronicles-1, chronicles-2, colossians, corinthians-1, corinthians-2, daniel, deuteronomy, ecclesiastes, ephesians, esther, exodus, ezekiel, ezra, galatians, genesis, habakkuk, haggai, hebrews, hosea, isaiah, james, jeremiah, job, joel, john, john-1, john-2, john-3, jonah, joshua, jude, judges, kings-1, kings-2, lamentations, leviticus, luke, malachi, mark, matthew, micah, nahum, nehemiah, numbers, obadiah, peter-1, peter-2, philemon, philippians, proverbs, psalms, revelation, romans, ruth, samuel-1, samuel-2, song-of-solomon, thessalonians-1, thessalonians-2, timothy-1, timothy-2, titus, zechariah, zephaniah

- **SHAK, all 38 of Shakespeare's plays**: a midsummer nights dream, alls well that ends well, antony and cleopatra, as you like it, coriolanus, cymbeline, hamlet, henry iv pt1, henry iv pt2, henry v, henry vi pt1, henry vi pt2, henry vi pt3, henry viii, julius caesar, king john, king lear, loves labors lost, macbeth, measure for measure, much ado about nothing, othello, pericles, richard ii, richard iii, romeo and juliet, taming of the shrew, the comedy of errors, the merchant of venice, the merry wives of windsor, the tempest, the two gentlemen of verona, the two noble kinsmen, the winters tale, timon of athens, titus andronicus, troilus and cressida, twelfth night

- **ABL: a collection of 70 great works of American and British Literature**[5]: adam bede, adonais, american scholar, an american slave, bartleby, being earnest, bleak house, caleb williams, david copperfield, defence of poetry, dorian gray, dracula, emma, felix holt, frankenstein, french revolution, great expectations, gullivers travels, heart of darkness, heaven and hell, huckleberry finn, innocence and experience, jane eyre, jude the obscure, eve of st agnes, ode on a grecian urn, ode to a nightingale, leaves of grass, little dorrit, lord jim, mansfield park, middlemarch, mlk i have a dream, moby dick, mohicans, mutual friend, northanger abbey, old curiosity shop, oliver twist, paradise lost, persuasion, pilgrims progress, portrait of a lady, pride and prejudice, red badge, return of the native, rights of woman, robinson crusoe, romola, sartor, scarlet letter, self reliance, sense and sensibility, seven gables, slave girl, tale of two cities, tess, the house of usher, the mill, the secret sharer, tom jones, tom sawyer, tristram shandy, uncle tom, us constitution, utopia, vanity fair, walden, washington square, wuthering heights

- **LAT, a collection of 329 great Latin works from the Perseus Digital Library**: see here (latin-works-metadata.jsonl) for a full list of all 329 works.

B Quality of Source-Derived Alignments

We use three different sets of alignments in our work – America's Public Bible (KJB-CA), JSTOR Understanding Series (KJB-JA, SHAK-JA, ABL-JA) and Passim (LAT-EJC). In this section, we discuss the text reuse models used for each collection and the quality of the detected alignments.

- **America's Public Bible (KJB-CA)**: We use alignments provided by Mullen (2016). As described in Mullen (2016, methods section), they first devise a set of features for each (Bible verse, newpaper page) pair to model their similarity: number of overlapping n-grams (with and without TF-IDF weighting), proportion of verse n-grams in newspaper, Wald–Wolfowitz runs test to test whether the positions of the matching tokens in the newspaper page are randomly scattered across the page or clustered together). Next, they sample a set of 1,700 potential (verse, newpaper page) matches and manually label each pair as a true match or not. They split the 1,700 pairs into train, dev and test sets and examine the performance of several models on this subset. They find that a neural network achieves the best performance, with an F1 of 0.92. They then use the trained neural network to label the remaining (verse, newspaper page) pairs in the collection.

- **JSTOR Understanding Series (KJB-JA, SHAK-JA, ABL-JA)**: We use alignments from the JSTOR Understanding Series (JSTOR Labs, 2019). For each source work, the JSTOR Labs team first generates a candidate set of derived JSTOR articles and chapters by performing a full-text search on JSTOR for the work's title, author, and main characters. Next, they extract all text appearing either in block quotes or between single- and double quotation-marks from each derived document. For each quote, they then identify the most similar subsequence in the source work (with

[5] The works of American and British Literature are selected as a subset of the American Literature (https://www.jstor.org/understand/american-literature) and British Literature (https://www.jstor.org/understand/british-literature) collections in the JSTOR Understanding Series. We select roughly the top 50% of works in each collection by total number of aligned derived works and combine them to make our single ABL dataset (70 total works). The dataset contains 20 works of American Literature and 50 works of British Literature (total collection sizes – American Literature (35), British Literature (98)). We do not include the bottom ≈ 50% of works in each collection since they are quoted in very few derived works (≈ 2-150 aligned derived works) and have extremely sparse labels.

the lowest Levenshtein distance). For each matched (source subsequence, derived quote) pair, they calculate a manually-designed confidence score $\in [0,1]$ based on the match size, match similarity (using Levenshtein distance) and signals indicating the presence of various text matches in surrounding derived text. Finally, examining a subset of their matched pairs, they perform a qualitative evaluation, finding that a confidence score threshold of 0.9 yielded the best balance of false positives to false negatives. We also confirm this finding qualitatively on a sampled set of alignments and use it in our work.

- **Passim (LAT-EJC)**: We use the Passim text alignment software (Smith et al., 2015) to detect quotes of the Latin texts in the JSTOR EJC. Passim uses the Smith–Waterman alignment algorithm to find all pairs of passages within longer documents (source and derived) with substantial alignments. Xu et al. (2014) quantitatively evaluated Passim on English text reuse in English documents, finding that it achieved pseudo-recall of roughly 0.9 and MAP of 0.2 - 0.5. In our work, we use the hyperparameter settings for Passim recommended by Xu et al. (2014). Manually examining our alignments, we find that detecting Latin text-reuse in English documents is easier than finding English text-reuse and confirm that Passim performs reasonably.

C Training lambdaMART: NDCG Formulation

There are two commonly used formulations of NDCG@k. One is the form we use to evaluate our models (§7.1), and the other is the formulation that lambdaMART is trained to optimize. For lambdaMART, $DCG_k = \sum_{i=1}^{k} \frac{2^{\text{label}_i} - 1}{log(i+1)}$. This formulation puts a stronger emphasis on retrieving highly relevant documents. This formulation is reasonable when passage labels are not large (e.g. $\in \{0,1,2,3\}$). However, in our datasets, some passages are quoted by a very large number of derived documents (e.g. 905 for a passage in Romeo & Juliet, 4949 for a passage in the Gospel of Matthew, 1536 for a passage in the U.S. Constitution). Due to these high counts, we opted for the non-exponentiated formulation of NDCG as we thought that it was more representative of model performance – e.g. if the most quoted passage in a document was quoted 500 times and the top ranked passage by the model was quoted 250 times, an NDCG@1 score of 0.5 (using the formulation we use) is much more representative of model performance than one of $\frac{2^{250}-1}{2^{500}-1} \approx 0$.

	Threshold (# quotes)		# Passages	
Dataset	Bottom	Top	Bottom	Top
KJB-CA	≤ 1	≥ 26	8,896	6,306
KJB-JA	≤ 0	≥ 3	12,834	7,937
SHAK-JA	≤ 0	≥ 3	52,479	31,868
ABL-JA	≤ 0	≥ 1	354,834	76,746
LAT-EJC	≤ 0	≥ 1	453,086	149,590

Table 8: Thresholds used to separate the highly quoted (top 20%) and least quoted (bottom 20%) passages in each dataset, and the corresponding number of passages in each group. Note, because most datasets contain many unquoted passages, the "bottom" group often contains more than 20% of the passages in the dataset.

To ensure that lambdaMART is trained to optimize an NDCG objective that is comparable to the one we use for evaluation, we log transform the original count labels of the passages in the training set $\text{label}_i = log_2(\text{count}_i + 1)$ before feeding them into lambdaMART. With this transformation, the numerators of the two DCG formulations are equivalent, but the denominators (the discount) differ slightly. We evaluate using the non-exponentiated version of NDCG (§7.1) with the original, untransformed quote counts on the test and dev sets.

D Quote Count Thresholds for Linguistic Analysis

In §5 we explore different linguistic features that affect a passage's quotability. For each dataset, we compare the highly quoted passages (top 20%) to the minimally-quoted ones (bottom 20%), testing if the feature values are significantly different between the two groups. Table 8 shows the thresholds used to identify the highly- and minimally-quoted passages in each dataset and the number of passages in each group. Note, because most datasets contain many unquoted passages, the groups of minimally quoted passages generally contain more than 20% of the total passages in the dataset (e.g. we split the entire dataset for ABL-JA and LAT-EJC with the bottom group consisting of all unquoted passages and the top group containing all passages with at least one quote).

E Features for Feature-based Models

For the feature-based model, each passage is featurized with three sets of features: bag-of-words, "quotability" features, and positional features.

E.1 Bag-of-words

We tokenize passages using Stanza (Qi et al., 2020). We use TFIDF-weighted bag of unigrams and bigrams as text features, keeping the top 100,000 that occur in at least five passages. We evaluate models with and without lemmatization. We use Stanza's 'ewt' package for English and 'perseus' package for Latin[6].

E.2 Quotability Features

Since Bendersky and Smith (2012) also explore applications of quotability identification models to literary works (Project Gutenberg), we primarily use their set of quotability features for our feature-based models:

- Length features: total number of words in the passage, total number of characters in the passage, average number of characters per word, minimum number of characters per word, maximum number of characters per word
- Capitalization: number of capital words in the passage
- Stop words: number of stop words in the passage, passage begins with a stop word
- Punctuation: Five binary features to indicate whether punctuation of type P is present in the passage, P = quotations, parentheses, colon, dash, semi-colon.
- Dialogue words: binary feature if the passage contains at least one common dialog term (English: say, says, said; Latin: list of 145 forms of the words 'dico' and 'loquor')
- Abstract concepts: Number of abstract concepts (e.g., adventure, charity, stupidity) in the passage. Following Bendersky and Smith (2012), we use a list of 176 abstract nouns available at www.englishbanana.com. We do not include this feature for the Latin dataset.
- Quantifiers: Total number of universal quantifiers in the passage (from a list of 20 quantifiers: 'all', 'always', 'each', 'entire', 'ever', 'every', 'everyone', 'full', 'fully', 'never', 'no', 'nobody', 'none', 'nothing', 'nowhere', 'total', 'totally', 'utterly', 'whole', 'wholy'). We do not include this feature for the Latin dataset.
- Emphasis: two binary features for if the passage contains a superlative adjective or a comparative adjective.
- Past participle: binary feature if passage contains a verb in past participle.
- Part of speech: two different sets of counts –
  - Five features for the number of occurrences of nouns, verbs, adjectives, adverbs, or pronouns in the passage.
  - Counts of part of speech triples, e.g. (DET, NOUN, VERB): we create a single feature for each unique part of speech triple and count the number of times it occurs in the passage. This is a slightly adapted version of POS triple feature from Bendersky and Smith (2012), who only include a limited number of POS triples based on calculations on their manually labeled validation set.
- Language Model: a log-likelihood ratio for the passage calculated as the ratio between log-likelihoods from a language model of quotable text and a background language model built on the entire source corpus. For the quotable text language model, we collect approximately 5,200 quotes on more than 200 subjects from the http://www.quotationspage.com/. This collection provides a diverse set of high-quality quotations on subjects ranging from Laziness and Genius to Technology and Taxes. This feature is only used for the English datasets.

Finally, we add two additional quotability features from Tan et al. (2018) and one from Danescu-Niculescu-Mizil et al. (2012):

- Personal Pronouns: three features for counts of 1st, 2nd, and 3rd person pronouns (Tan et al., 2018).

[6] https://stanfordnlp.github.io/stanza/available_models.html

- Generality: number of indefinite articles, only for English dataset (Tan et al., 2018).

- Verb tenses: three features for counts of past, present and future tense verbs (Danescu-Niculescu-Mizil et al., 2012).

E.3 Positional Features

Since, as noted by Tan et al. (2018), a passage's position in a source document is an important feature for determining its quotability, we include the following positional features for each passage (each source dataset has unique positional features). Features such as "Verse index in (chapter, entire book)" with parentheticals indicate that we create multiple features, here one for the verse index in the chapter and the other for the verse index across the entire book.

- **King James Bible**:
  - Verse index in (chapter, entire book)
  - Chapter index in book
  - Relative (0 to 1) verse position in (chapter, book), both raw decimal and one-hot vectors by decile
  - Is (first, last) verse of (book, chapter)
  - Is (first, last) chapter of book

- **Shakespeare**:
  - Line index in (scene, act, play)
  - Scene index in (act, play)
  - Act index in play
  - Relative (0 to 1) line position in (scene, act, play), both raw decimal and one-hot vectors by decile
  - Relative (0 to 1) scene position in (act, play), both raw decimal and one-hot vectors by decile
  - Relative (0 to 1) act position in play, both raw decimal and one-hot vectors by decile
  - Is (first, last) line of (scene, act, play)
  - Is (first, last) scene of (act, play)
  - Is (first, last) act of play

- **American & British Literature**:
  - Sentence index in book
  - Paragraph index in book
  - Chapter index in book
  - Relative (0 to 1) (sentence, paragraph, chapter) position in book, both raw decimal and one-hot vectors by decile.
  - Is (first, last) sentence of (paragraph, chapter, doc)
  - Is (first, last) paragraph of (chapter, book)
  - Is (first, last) chapter of book

- **Latin**: Each document in the Perseus library is split up into sections which roughly break up the text by line/paragraph/sentence etc. depending on the genre and specific work. We use Stanza's Latin models to break up the text into sentences, but use this section boundary information for positional data, as seen below.
  - Sentence index in book
  - (Start, end) section index in book (sentences can span multiple sections)
  - Relative (0 to 1) (sentence, start section, end section) position in book, both raw decimal and one-hot vectors by decile.
  - Is (first, last) sentence of (book, start section, end section)
  - Is (first, last) (start, end) section of book

F Special Positional Tokens for BERT-based Models

For the BERT-based models, we add special tokens to each passage to act as positional indicators so that the model has a better sense of which part of the document it is reading (Alberti et al., 2019). Just as for the feature-based models (§E.3), the positional tokens vary across datasets.

- **King James Bible**:
  - [`Starts-Book`]: added to the beginning of the first verse in the entire document (e.g. Luke 1.1)
  - [`Starts-Chapter`]: added to the beginning of the first verse in each chapter (e.g. Luke 1.1, 2.1)
  - [`Ends-Book`]: added to the end of the last verse in the entire document (e.g. Luke 24.53)
  - [`Ends-Chapter`]: added to the end of the last verse in each chapter (e.g. Luke 1.80, 2.52)

  - [Book@n]: added to the beginning of each verse indicating its relative position in the entire document. n ranges from 0-9 indicating in which decile in the document the verse occurs.
  - [Chapter@n]: added to the beginning of each verse indicating its relative position in the chapter. n ranges from 0-9 indicating which decile in the chapter the verse occurs.

- **Shakespeare**:
  - [Starts-Play]: added to the beginning of the first line in the entire play
  - [Starts-Act]: added to the beginning of the first line in each act
  - [Starts-Scene]: added to the beginning of the first line in each scene
  - [Ends-Play]: added to the end of the last line in the entire play
  - [Ends-Act]: added to the end of the last line in each act
  - [Ends-Scene]: added to the end of the last line in each scene
  - [Play@n]: added to the beginning of each line indicating its relative position in the entire play. n ranges from 0-9 indicating in which decile in the play the verse occurs.
  - [Act@n]: added to the beginning of each line indicating its relative position in the act. n ranges from 0-9 indicating in which decile in the act the verse occurs.
  - [Scene@n]: added to the beginning of each line indicating its relative position in the scene. n ranges from 0-9 indicating in which decile in the scene the verse occurs.

- **American & British Literature**:
  - [Starts-Book]: added to the beginning of the first sentence in the entire document
  - [Starts-Chapter]: added to the beginning of the first sentence in each chapter
  - [Starts-Paragraph]: added to the beginning of the first sentence in each paragraph
  - [Ends-Book]: added to the end of the last sentence in the entire document
  - [Ends-Chapter]: added to the end of the last sentence in each chapter
  - [Ends-Paragraph]: added to the ends of the last sentence in each paragraph
  - [Book@n]: added to the beginning of each sentence indicating its relative position in the entire document. n ranges from 0-9 indicating in which decile in the document the verse occurs.
  - [Chapter@n]: added to the beginning of each sentence indicating the relative position of its chapter in the entire document. n ranges from 0-9 indicating in which decile in the document the chapter occurs.
  - [Paragraph@n]: added to the beginning of each sentence indicating the relative position of its paragraph in the entire document. n ranges from 0-9 indicating in which decile in the document the paragraph occurs.

- **Latin**: As noted in §E.3, each document in the Perseus library is split up into sections which roughly break up the text by line/paragraph/sentence etc. depending on the genre and specific work. We use Stanza's Latin models to break up the text into sentences, but use this section boundary information for positional data, as seen below.
  - [Starts-Book]: added to the beginning of the first sentence in the entire document
  - [Ends-Book]: added to the end of the last sentence in the entire document
  - [Starts-Section]: inserted wherever a section (according to Perseus) begins. This can be anywhere in the sentence, not necessarily just in the beginning or end.
  - [Ends-Section]: inserted wherever a section (according to Perseus) ends. This can be anywhere in the sentence, not necessarily just in the beginning or end.
  - [Book@n]: added to the beginning of each sentence indicating its relative position in the entire document. n ranges from 0-9 indicating in which decile in the document the verse occurs.

G Sequence length limits: BERT-based models

For the BERT-based models which operate on single passages (RoBERTA$_{single}$, RoBERTA$_{pair}$, XLM-R$_{single}$, XLM-R$_{pair}$), we cap passages at the following predetermined lengths:

- SHAKE: 100 WordPieces ($>$ 99% of passages)
- ABL: 200 WordPieces ($>$ 99% of passages)
- LAT: 200 WordPieces ($>$ 99% of passages)
- KJB: no cap, all sequences under 113 WordPieces

For the BERT-based models that operate on sequences of multiple sentences (RoBERTA$_{seq}$, XLM-R$_{seq}$), we create an example by greedily adding passages until we hit the cap of 512 WordPieces. If any single sequence is longer than 512 WordPieces, it is capped at that length.

H Searched Hyperparameters & Best Model Configurations

For the feature-based models, we use TFIDF-weighted bag-of-unigrams and bigrams as text features, keeping the top 100k that occur in at least five passages. We evaluate models with and without lemmatization. For poisson regression and SVM^{rank}, we search over regularization parameters in 10^x where $x \in \{$-5, -4, -3, -2, -1, 0, 1$\}$. For SVM^{rank} we also search over the number, n, of negative passages to sample per positive passage, $n \in \{1, 3, 5, 10\}$. We create a separate training example for each (positive, negative) passage pair. For lambdaMART we search over learning rate $\in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, max tree depth $\in \{4, 6, 8, 10\}$ and number of boosting rounds $\in \{50, 100, 250, 500\}$.

For the neural models, we use Adam (Kingma and Ba, 2014) and search over learning rate $\in \{2e\text{-}5, 3e\text{-}5, 5e\text{-}5\}$ and batch size $\in \{16, 32\}$. We use dropout with probability 0.1. We train single sentence models for up to 10 epochs and sequential sentence models for up to 20, evaluating on the validation set after each epoch and selecting the highest performing model. For the pairwise models, we search over the number, n, of negative passages to sample per positive passage, $n \in \{1, 3, 5, 10\}$ and create a separate training example for each pair.

		Lemmatize	α
KJB-CA	Fold 1	True	1e-3
	Fold 2	True	1e-3
	Fold 3	False	1e-5
	Fold 4	True	1e-3
	Fold 5	True	1e-3
KJB-JA	Fold 1	True	1e-3
	Fold 2	True	1e-4
	Fold 3	True	1e-4
	Fold 4	True	1e-4
	Fold 5	False	1e-4
SHAKE-JA	Fold 1	False	1e-5
	Fold 2	True	1e-4
	Fold 3	True	1.0
	Fold 4	True	0.1
	Fold 5	True	1e-3
ABL-JA	Fold 1	True	1e-4
	Fold 2	True	1e-4
	Fold 3	True	1e-4
	Fold 4	True	1e-3
	Fold 5	True	1e-4
LAT-EJC	Fold 1	False	1e-5
	Fold 2	False	1e-5
	Fold 3	True	1e-5
	Fold 4	False	1e-5
	Fold 5	False	1e-5

Table 9: Hyperparameters of the best performing **poisson regression** models for all five datasets. Since we perform 5-fold cross validation, there are separate best hyperparameters for each fold. Lemmatize: if tokens are lemmatized, α: regularization parameter.

Tables 9, 10, and 11 lists the best hyperparameter configurations for the poisson regression, SVM^{rank}, and lambdaMART models, respectively, across all datasets and folds. We use scikit-learn's (Pedregosa et al., 2011) implementations of poisson regression and SVM^{rank}. We use XGBoost's (Chen and Guestrin, 2016) implementation of lambdaMART.

Table 12 lists the best hyperparameter configurations for the different neural models across all datasets and folds. We train models on a single Nvidia V100 GPU (32GB configuration). We train models with Pytorch (Paszke et al., 2019) and use the pretrained RoBERTA and XLM-R models from the Huggingface Transformers library (Wolf et al., 2019).

		Lemmatize	C	# Negative
KJB-CA	Fold 1	True	0.1	3
	Fold 2	False	1e-2	10
	Fold 3	False	0.1	1
	Fold 4	False	0.1	5
	Fold 5	False	0.1	3
KJB-JA	Fold 1	True	1e-2	5
	Fold 2	True	1e-3	10
	Fold 3	False	1e-3	10
	Fold 4	True	1e-2	3
	Fold 5	True	1e-3	10
SHAKE-JA	Fold 1	True	1e-2	5
	Fold 2	False	1e-2	5
	Fold 3	True	0.1	3
	Fold 4	True	1e-2	5
	Fold 5	False	1e-2	10
ABL-JA	Fold 1	True	1e-2	3
	Fold 2	True	1e-2	5
	Fold 3	True	1e-2	3
	Fold 4	False	1e-2	10
	Fold 5	False	1e-3	10
LAT-EJC	Fold 1	True	1e-2	10
	Fold 2	True	1e-2	3
	Fold 3	True	1e-2	10
	Fold 4	True	1e-3	10
	Fold 5	True	1e-2	10

Table 10: Hyperparameters of the best performing **SVM**$^{\mathbf{rank}}$ models for all five datasets. Since we perform 5-fold cross validation, there are separate best hyperparameters for each fold. Lemmatize: if tokens are lemmatized, C: regularization parameter, # Negative: number of negative samples per positive passage.

		Lemmatize	LR	MTD	# BR
KJB-CA	Fold 1	False	0.4	8	250
	Fold 2	False	0.4	4	100
	Fold 3	True	0.5	10	100
	Fold 4	False	0.2	4	250
	Fold 5	True	0.1	10	500
KJB-JA	Fold 1	True	0.4	10	50
	Fold 2	True	0.5	8	50
	Fold 3	False	0.1	8	100
	Fold 4	True	0.3	10	50
	Fold 5	False	0.4	4	50
SHAKE-JA	Fold 1	True	0.01	8	250
	Fold 2	True	0.1	6	50
	Fold 3	False	0.4	6	100
	Fold 4	False	0.4	6	100
	Fold 5	False	0.4	6	100
ABL-JA	Fold 1	True	0.3	4	250
	Fold 2	False	0.2	10	250
	Fold 3	True	0.5	4	100
	Fold 4	False	0.4	10	250
	Fold 5	True	0.3	10	250
LAT-EJC	Fold 1	True	0.4	8	250
	Fold 2	True	0.2	8	500
	Fold 3	True	0.1	8	500
	Fold 4	True	0.2	8	500
	Fold 5	True	0.4	8	250

Table 11: Hyperparameters of the best performing **lambdaMART** models for all five datasets. Since we perform 5-fold cross validation, there are separate best hyperparameters for each fold. Lemmatize: if tokens are lemmatized, LR: learning rate, MTD: maximum tree depth, # BR: number of boosting rounds.

		Fold 1				Fold 2				Fold 3				Fold 4				Fold 5			
		E	LR	BS	Neg	E	LR	BS	Neg	E	LR	BS	Neg	E	LR	BS	Neg	E	LR	BS	Neg
KJB-CA	RoBERTA$_{single}$	9	3e-5	16	–	6	2e-5	32	–	8	2e-5	16	–	8	2e-5	32	–	8	3e-5	32	–
	RoBERTA$_{pair}$	9	2e-5	16	1	10	3e-5	32	3	10	2e-5	16	10	7	2e-5	32	1	9	2e-5	32	10
	RoBERTA$_{seq}$	6	3e-5	16	–	14	3e-5	16	–	20	2e-5	16	–	17	3e-5	32	–	12	3e-5	16	–
KJB-JA	RoBERTA$_{single}$	9	3e-5	32	–	10	5e-5	32	–	4	2e-5	32	–	10	3e-5	16	–	3	3e-5	16	–
	RoBERTA$_{pair}$	7	2e-5	16	10	7	3e-5	32	10	9	2e-5	16	10	3	2e-5	32	3	8	2e-5	32	10
	RoBERTA$_{seq}$	16	5e-5	32	–	11	5e-5	16	–	20	5e-5	32	–	14	5e-5	32	–	13	3e-5	16	–
SHAKE-JA	RoBERTA$_{single}$	6	2e-5	16	–	4	2e-5	32	–	2	2e-5	32	–	3	3e-5	32	–	7	2e-5	32	–
	RoBERTA$_{pair}$	8	2e-5	32	5	4	2e-5	16	3	7	2e-5	16	5	5	3e-5	16	1	9	3e-5	32	10
	RoBERTA$_{seq}$	2	5e-5	32	–	8	5e-5	16	–	14	5e-5	32	–	7	5e-5	16	–	11	5e-5	32	–
ABL-JA	RoBERTA$_{single}$	2	2e-5	16	–	1	2e-5	16	–	3	3e-5	32	–	10	2e-5	16	–	2	2e-5	16	–
	RoBERTA$_{pair}$	7	3e-5	32	1	5	5e-5	32	5	2	5e-5	32	10	4	2e-5	16	3	1	5e-5	32	10
	RoBERTA$_{seq}$	3	3e-5	32	–	3	2e-5	32	–	5	5e-5	32	–	2	2e-5	32	–	4	5e-5	32	–
LAT-EJC	XLM-R$_{single}$	6	2e-5	32	–	9	3e-5	32	–	9	2e-5	32	–	8	3e-5	32	–	4	2e-5	16	–
	XLM-R$_{pair}$	9	2e-5	32	3	3	3e-5	32	10	3	2e-5	32	3	4	3e-5	32	1	3	2e-5	32	5
	XLM-R$_{seq}$	10	3e-5	32	–	11	2e-5	16	–	10	2e-5	16	–	12	3e-5	32	–	16	2e-5	16	–

Table 12: Hyperparameters of the best performing **neural models** for all five datasets. Since we perform 5-fold cross validation, there are separate best hyperparameters for each fold. E: number of epochs, LR: learning rate, BS: batch size, Neg: number of negative passages sampled per positive passage (only for pairwise models).