Catching Attention with Automatic Pull Quote Selection

Tanner Bohn
Western University
London, ON, Canada
tbohn@uwo.ca

Charles X. Ling
Western University
London, ON, Canada
charles.ling@uwo.ca

Abstract

To advance understanding on how to engage readers, we advocate the novel task of automatic pull quote selection. Pull quotes are a component of articles specifically designed to catch the attention of readers with spans of text selected from the article and given more salient presentation. This task differs from related tasks such as summarization and clickbait identification by several aspects. We establish a spectrum of baseline approaches to the task, ranging from handcrafted features to a neural mixture-of-experts to cross-task models. By examining the contributions of individual features and embedding dimensions from these models, we uncover unexpected properties of pull quotes to help answer the important question of what engages readers. Human evaluation also supports the uniqueness of this task and the suitability of our selection models. The benefits of exploring this problem further are clear: pull quotes increase enjoyment and readability, shape reader perceptions, and facilitate learning. Code to reproduce this work is available at https://github.com/tannerbohn/AutomaticPullQuoteSelection.

1 Introduction

Discovering what keeps readers engaged is an important problem. We thus propose the novel task of automatic pull quote (PQ) selection accompanied with a new dataset and insightful analysis of several motivated baselines. PQs are graphical elements of articles with thought provoking spans of text pulled from an article by a writer or copy editor and presented on the page in a more salient manner (French, 2018), such as in Figure 1.

> In a way, a PQ is like clickbait, except that it is not lying to people.

Figure 1: A pull quote from this paper chosen with the help of our best performing model (see Section 5.3).

PQs serve many purposes. They provide temptation (with unusual or intriguing phrases, they make strong entrypoints for a browsing reader), emphasis (by reinforcing particular aspects of the article), and improve overall visual balance and excitement (Stovall, 1997; Holmes, 2015). PQ frequency in reading material is also significantly related to information recall and student ratings of enjoyment, readability, and attractiveness (Wanta and Gao, 1994; Wanta and Remy, 1994).

The problem of automatically selecting PQs is related to the previously studied tasks of headline success prediction (Piotrkowicz et al., 2017; Lamprinidis et al., 2018), clickbait identification (Potthast et al., 2016; Chakraborty et al., 2016; Venneti and Alam, 2018), as well as key phrase extraction (Hasan and Ng, 2014) and document summarization (Nenkova and McKeown, 2012). However, in Sections 5.4 and 5.5 we provide experimental evidence that performing well on these previous tasks does not translate to performing well at PQ selection. Each of these types of text has a different function in the context of engaging a reader. The title tells the reader what the article is about and sets the tone. Clickbait makes

Proceedings of the 28th International Conference on Computational Linguistics, pages 62–76
Barcelona, Spain (Online), December 8-13, 2020

unwarranted enticing promises of what the article is about. Key phrases and summaries help the reader decide whether the topic is of interest. And PQs provide *specific* intriguing entrypoints for the reader or can *maintain* interest once reading has begun by providing glimpses of interesting things to come. With their unique qualities, we believe PQs satisfy important roles missed by these popular existing tasks.

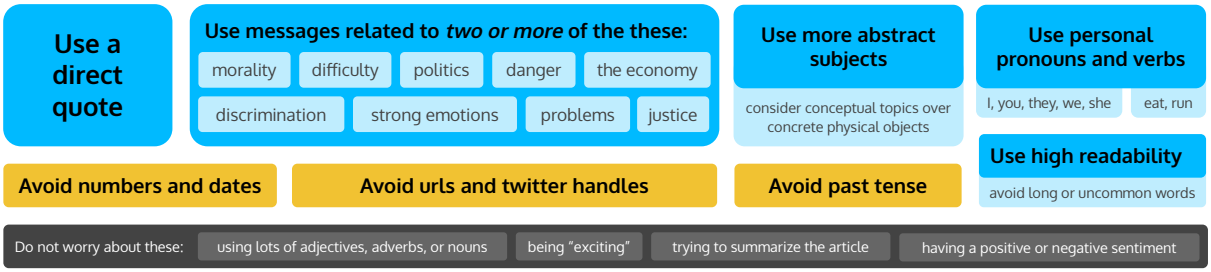


Figure 2: Factors suggested by our results to be important (and unimportant) in creating pull quotes.

In this work we define PQ selection as a sentence classification task and create a dataset of articles and their expert-selected PQs from a variety of news sources. We establish a number of approaches with which to solve and gain insight into this task: (1) handcrafted features, (2) n-gram encodings, (3) Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) embeddings combined with a progression of neural architectures, and (4) cross-task models. Via each of these model groups, we uncover interesting patterns (summarized in Figure 2). For example, among handcrafted features, sentiment and arousal are surprisingly uninformative features, overshadowed by presence of quotation marks and reading difficulty. Analysing individual SBERT embedding dimensions also helps understand the particular themes that make for a good PQ. We also find that combining SBERT sentence and document embeddings in a mixture-of-experts manner provide the best performance at PQ selection. The suitability of our models at PQ selection is also supported via human evaluation.

The main contributions are:

1. We describe several motivated approaches for the new task of PQ selection, including a mixture-of-experts approach to combine sentence and document embeddings (Section 3).

2. We construct a dataset for training and evaluation of automatic PQ selection (Section 4).

3. We inspect the performance of our approaches to gain a deeper understanding of PQs, their relation to other tasks, and what engages readers (Section 5). Figure. 2 summarizes these findings.

2 Related Work

In this section, we look at three natural language processing tasks related to PQ selection: (1) headline quality prediction, (2) clickbait identification, and (3) summarization and keyphrase extraction. These topics motivate the cross-task models whose performance on PQ selection is reported in Section 5.4.

2.1 Headline Quality Prediction

When a reader comes across a news article, the headline is often the first thing given a chance to catch their attention, thus predicting their success is a strongly motivated task. Once a reader decides to check out the article, it is up to the content (including PQs) to maintain their engagement.

In (Piotrkowicz et al., 2017), the authors experimented with two sets of features: journalism-inspired (which aim to measure how news-worthy the topic itself is), and linguistic style features (reflecting properties such as length, readability, and parts-of-speech – we consider such features here). They found that overall the simpler style features work better than the more complex journalism-inspired features at predicting social media popularity of news articles. The success of simple features is also reflected in (Lamprinidis et al., 2018), which proposed multi-task training of a recurrent neural network to not only predict headline popularity given pre-trained word embeddings, but also predict its topic and parts-of-speech tags. They found that while the multi-task learning helped, it performed only as well as a logistic

regression model using character n-grams. Similar to these previous works, we also evaluate several expert-knowledge based features and n-grams, however, we expand upon this to include a larger variety of models and provide a more thorough inspection of performance to understand what engages readers.

2.2 Clickbait Identification

The detection of a certain type of headline – clickbait – is a recently popular task of study. Clickbait is a particularly catchy headline and form of false advertising used by news outlets which lure potential readers but often fail to meet expectations, leaving readers disappointed (Potthast et al., 2016). Clickbait examples include "You Won't Believe..." or "X Things You Should...". We suspect that the task of distinguishing between clickbait and non-clickbait headlines is related to PQ selection because both tasks may rely on identifying the catchiness of a span of text. However, PQs attract your attention with content truly in the article. In a way, a PQ is like clickbait, except that it is not lying to people.

In (Venneti and Alam, 2018), the authors found that measures of topic novelty (estimated using LDA) and surprise (based on word bi-gram frequency) were strong features for detecting clickbait. In our work however, we investigate the interesting topics themselves (Section 5.3). A set of 215 handcrafted features were considered in (Potthast et al., 2016) including sentiment, length statistics, specific word occurrences, but the authors found that the most successful features were character and word n-grams. The strength of n-gram features at this task is also supported by (Chakraborty et al., 2016). While we also demonstrate the surprising effectiveness of n-grams and consider a variety of handcrafted features for our particular task, we examine more advanced approaches that exhibit superior performance.

2.3 Summarization and Keyphrase Extraction

Document summarization and keyphrase extraction are two well-studied NLP tasks with the goals of capturing and conveying the main topics and key information discussed in a body of text (Turney, 1999; Nenkova and McKeown, 2012). Keyphrase extraction is concerned with doing this at the level of individual phrases, while extractive document summarization (which is just one type of summarization (Nenkova et al., 2011)) aims to do this at the sentence level. Approaches to summarization have roughly evolved from unsupervised extractive heuristic-based methods (Luhn, 1958; Mihalcea and Tarau, 2004; Erkan and Radev, 2004; Nenkova and Vanderwende, 2005; Haghighi and Vanderwende, 2009), to supervised and often abstractive deep-learning approaches (Nallapati et al., 2016b; Nallapati et al., 2016a; Nallapati et al., 2017; Zhang et al., 2019). Approaches to keyphrase extraction fall into similar groups, with unsupervised approaches including (Tomokiyo and Hurst, 2003; Mihalcea and Tarau, 2004; Liu et al., 2009), and supervised approaches including (Turney, 1999; Medelyan et al., 2009; Romary, 2010).

While summarization and keyphrase extraction are concerned with what is *important* or representative in a document, we instead are interested in understanding what is *engaging*. While these two concepts may seem very similar, in Sections 5.4 and 5.4 we provide evidence of their difference by demonstrating that what makes for a good summary does not make for a good PQ.

3 Models

We consider four groups of approaches for the PQ selection task: (1) handcrafted features (Section 3.1), (2) n-gram features (Section 3.2), (3) SBERT embeddings combined with a progression of neural architectures (Section 3.3), and (4) cross-task models (Section 3.4). As discussed further in Section 4, these approaches aim to determine the probability that a given article sentence will be used for a PQ.

3.1 Handcrafted Features

Our handcrafted features can be loosely grouped into three categories: surface, parts-of-speech, and affect, each of which we will provide justification for. For the classifier we will use AdaBoost (Hastie et al., 2009) with a decision tree base estimator, as this was found to outperform simpler classifiers without requiring much hyperparameter tuning.

3.1.1 Surface Features

- **Length**: We expect that writers have a preference to choose PQs which are concise. To measure length, we will use the total character length, as this more accurately reflects the space used by the text than the number of words.

- **Sentence position**: We consider the location of the sentence in the document (from 0 to 1). This is motivated by the finding in summarization that summary-suitable sentences tend to occur near the beginning (Braddock, 1974) – perhaps a similar trend exists for PQs.

- **Quotation marks**: We observe that PQs often contain content from direct quotations. As a feature, we thus include the count of opening and closing double quotation marks.

- **Readability**: Motivated by the assumption that writers will not purposefully choose difficult-to-read PQs, we consider two readability metric features: (1) **Flesch Reading Ease**: This measure (R_{Flesch}) defines reading ease in terms of the number of words per sentence and the number of syllables per word (Flesch, 1979). (2) **Difficult words**: This measure ($R_{difficult}$) is the percentage of unique words which are considered "difficult" (at least six characters long and not in a list of $\sim$3000 easy-to-understand words). See Appendix A for details.

3.1.2 Part-of-Speech Features

We include the word density of part-of-speech (POS) tags in a sentence as a feature. As suggested by (Piotrkowicz et al., 2017) with respect to writing good headlines, we suspect that verb (VB) and adverb (RB) density will be informative. We also report results on the following: cardinal digit (CD), adjective (JJ), modal verb (MD), singular noun (NN), proper noun (NNP), personal pronoun (PRP).

3.1.3 Affect Features

Events or images that are shocking, filled with emotion, or otherwise exciting will attract attention (Schupp et al., 2007). However, this does not necessarily mean that text describing these things will catch reader interest as reliably (Aquino and Arnell, 2007). To determine how predictive sentence affect properties are of PQ suitability, we include the following features:

Positive sentiment (A_{pos}) and **negative sentiment**(A_{neg}).

Compound sentiment ($A_{compound}$). This combines the positive and negative sentiments to represent overall sentiment between -1 and 1.

Valence ($A_{valence}$) and **arousal** ($A_{arousal}$): Valence refers to the pleasantness of a stimulus and arousal refers to the intensity of emotion provoked by a stimulus (Warriner et al., 2013). In (Aquino and Arnell, 2007), the authors specifically note that it is the arousal level of words, and not valence which is predictive of their effect on attention (measured via reaction time). Measuring early cortical responses and recall, (Kissler et al., 2007) observed that words of greater valence were both more salient and memorable. To measure valence and arousal of a sentence, we use the averaged word rating, utilizing word ratings from the database introduced by (Warriner et al., 2013).

Concreteness ($A_{concreteness}$): This is "the degree to which the concept denoted by a word refers to a perceptible entity" (Brysbaert et al., 2014). As demonstrated by (Sadoski et al., 2000), concrete texts are better recalled than abstract ones and concreteness is a strong predictor of text comprehensibility, interest, and recall. To measure concreteness of a sentence, we use the averaged word rating, utilizing word ratings in the database introduced by (Brysbaert et al., 2014).

3.2 N-Gram Features

We consider character-level and word-level n-gram text representations, shown to perform well in related tasks (Potthast et al., 2016; Chakraborty et al., 2016; Lamprinidis et al., 2018). A passage of text is then represented by a vector of the counts of the individual n-grams it contains. We use a logistic regression classifier with these representations.

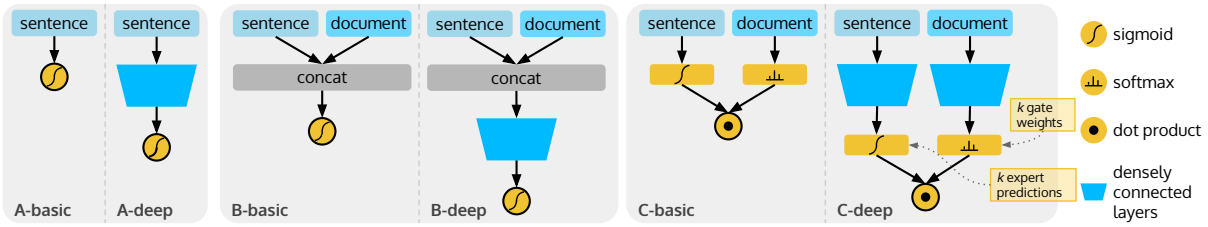


Figure 3: The progression of neural network architectures combined with SBERT sentence and document embeddings. Group A only uses sentence embeddings, while groups B and C also use document embeddings. In group C, they are combined in a mixture-of-experts fashion (the width of the sigmoid and softmax layers is equal to the # experts). For each group, there is a basic version and deep version.

3.3 SBERT Embeddings with a Progression of Neural Architectures

All other models described in this work use only the single sentence to predict PQ probability. To understand the importance of considering the entire article when choosing PQs, we consider three groups of neural architectures, as shown in Figure 3.

Group A. These neural networks only take the sentence embedding as input. In the **A-basic** model, there are no hidden layers. In **A-deep**, the embedding passes through a set of densely connected layers.

Group B. These models receive the sentence embedding and a whole-document embedding as input. This allows the models to account for document-dependent patterns. These embeddings are concatenated and connected to the output node (**B-basic**), or first pass through densely connected layers (**B-deep**).

Group C. These networks also receive sentence and document embeddings, but they are combined in a mixture-of-experts manner (Jacobs et al., 1991). That is, multiple predictions are produced by a set of "experts" and a gating mechanism determines the weighting of these predictions for a given input. The motivation is that there may be many "types" of articles, each requiring paying attention to different properties when choosing a PQ. If each of k experts generates a prediction, we can use the document embedding to determine the weighting over the predictions. In Figure 3c, k corresponds to the width of the sigmoid and softmax layers, which are then combined with a dot product to produce the final prediction. In **C-deep**, the embeddings first pass through a set of densely connected layers (non-shared weights) as shown in the right of Figure 3c, while in **C-basic**, they do not.

To embed sentences and documents, we make use of a pre-trained Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019). SBERT is a modification of BERT (Bidirectional Encoder Representations from Transformers) – a language representation model which performs well on a wide variety of tasks (Devlin et al., 2018). SBERT is designed to more efficiently produce semantically meaningful embeddings (Reimers and Gurevych, 2019). We computed document embeddings by averaging SBERT sentence embeddings.

3.4 Cross-Task Models

To test the similarity of PQ selection with related tasks , we use the following models: **Headline popularity**: We train a model to predict the popularity of a headline (using SBERT embeddings and linear regression) with the dataset introduced by (Moniz and Torgo, 2018). This dataset includes feedback metrics for about 100K news articles from various social media platforms. We apply this model to PQ selection by predicting the popularity of each sentence, scaling the predictions for each article to lie in $[0, 1]$ and interpreting these values as PQ probability. **Clickbait identification**: We train a model to discriminate between clickbait and non-clickbait headlines (using SBERT embeddings and logistic regression) with the dataset introduced by (Chakraborty et al., 2016). Clickbait probability is used as a proxy for PQ probability. **Summarization**: Using a variety of extractive summarizers, we score each sentence in an article, scale the values to lie in $[0, 1]$, and interpret these values as PQ probability. No training is required for this model. Appendix. A contain implementation details of these models

4 Experimental Setup

To support the new task of automatic PQ selection, we both construct a new dataset and describe a suitable evaluation metric.

4.1 Datatset Construction

To conduct our experiments, we create a dataset using articles from several online news outlets: National Post, The Intercept, Ottawa Citizen, and Cosmopolitan. For each outlet, we identify those articles containing at least one pull quote. From these articles, we extract the *body*, *edited PQs*, and *PQ source sentences*. The body contains the full list of sentences composing the body of the article. The edited PQs are the pulled texts as they appear after being augmented by the editor to appear as pull quotes[1]. The PQ source sentences are the article sentences from which the edited PQs came. In this work, we aim to determine whether a given article sentence is a source sentence or not[2].

Dataset statistics are reoprted in Table 1. It contains ∼27K positive samples (PQ source sentences—which we simply call PQ sentences) and ∼680K negative samples (non-PQ sentences). The positive to negative ratio is 1:26 (taken into consideration when training our classifiers with balanced class weights). For all experiments, we use the same training/validation/test split of the articles (70/10/20).

	nationalpost	theintercept	ottawacitizen	cosmopolitan	train	val	test	all
# articles	11112	1183	1066	1267	10239	1462	2927	14628
# PQ	16307	2671	1087	2360	15709	2235	4481	22425
# PQ/article	1.47	2.26	1.02	1.86	1.53	1.53	1.53	1.53
# sentences/PQ	1.16	1.23	1.32	1.24	1.19	1.18	1.19	1.19
# sentences/article	40.49	97.94	38.35	79.03	48.47	47.8	48.06	48.32
# pos samples	18975	3274	1436	2906	18640	2625	5326	26591
# neg samples	430959	112588	39443	97230	477609	67258	135353	680220

Table 1: Statistics of our PQ dataset, composed of articles from four different news outlets. Only articles with at least one PQ are included in the dataset.

4.2 Evaluation

What do we want to measure? We want to evaluate a PQ selection model on its ability to determine which sentences are more likely to be chosen by an expert as PQ source sentences.

Metric. We will use the probability that a random PQ source sentence is scored by the model above a random non-source sentence from the same article (i.e. AUC). Let $a_{inclusions}$ be the binary vector indicating whether each sentence of article a is truly a PQ source sentence, and let $\hat{a}_{inclusions}$ be the corresponding predicted probabilities. Our metric can then be computed with Equation 1, which computes the AUC averaged across articles.

$$AUC_{avg} = \frac{1}{\#articles} \sum_{a \in articles} AUC(a_{inclusions}, \hat{a}_{inclusions}) \quad (1)$$

Why average across articles? By averaging scores for each article instead of for all sentences at the same time, the evaluation method accounts for the observation that some articles may be more "pull-quotable" than others. If articles are instead combined when computing AUC, an average sentence from an interesting article can be ranked higher than the best sentence from a less interesting article.

5 Experimental Results

We present our experimental results and analysis for the four groups of approaches: handcrafted features (Section 5.1), n-gram features (Section 5.2), SBERT embeddings combined with a progression of

[1]This can include replacing pronouns such as "she", "they", "it", with the more precise nouns or proper nouns, or shortening sentences by removing individual words or clauses, or even replacing words with ones of a similar meaning but different length in order to achieve a clean text rag.

[2]A PQ source sentence could be only part of a multi-sentence PQ or contain the PQ inside it.

neural architectures (Section 5.3), and cross-task models (Section 5.4). We also perform human evaluation of several models (Section 5.5). Appendix A contains implementation details of our models, and Appendix C includes examples of PQ sentences selected by several models on various articles.

5.1 Handcrafted Features

The performance of each of our handcrafted features is provided in Figure 4a. There are several interesting observations, including some that support and contradict hypotheses made in Section 3.1:

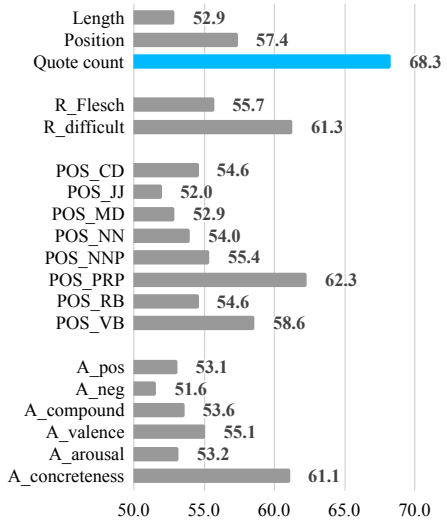
Sentence position. Simply using the sentence position works better than random guessing. When we inspect the distribution of this feature value for PQ and non-PQ sentences in Figure 4b, we see that PQ sentences are not uniformly distributed throughout articles, but rather tend to occur slightly more often around a quarter of the way through the article.

Quotation mark count.. The number of quotation marks is by far the best feature in this group, confirming that direct quotations make for good PQs. We find that a given non-PQ sentence is ∼3 times more likely not to contain quotation marks than a PQ sentence.

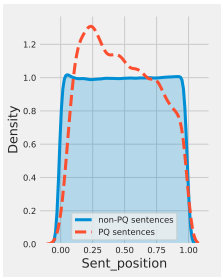
Reading difficulty. The fraction of difficult words is the third-best handcrafted feature, outperforming the Flesch metric. As suggested in Section 3.1.1 we find that PQ sentences are indeed easier to read than non-PQ sentences.

POS tags. Of the POS tag densities, personal pronoun (PRP) and verb (VB) density are the most informative. Inspecting the feature distributions, we see that PQs tend to have slightly higher PRP density as well as VB density – suggesting that sentences about people doing things are good candidates for PQs.

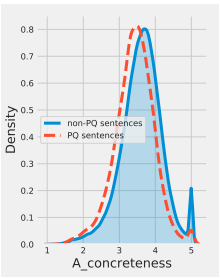
Affect features. Affect features tended to perform poorly, contradicting our intuition that more exciting or emotional sentences would be chosen for PQs. However, concreteness is indeed an informative feature, with *decreased* concreteness unexpectedly being better (see Figure 4c). Given the memorability that comes with more concrete texts (Sadoski et al., 2000), this suggests that something else may be at work in order to explain the beneficial effects of PQs on learning outcomes (Wanta and Gao, 1994; Wanta and Remy, 1994).



(a) Performance of handcrafted features



(b) Sentence position



(c) Concreteness

Figure 4: The value distributions for two interesting handcrafted features for both non-PQ sentences (solid blue region) and PQ sentences (dashed orange lines).

5.2 N-Gram Features

The results for our n-gram models are provided in Table 2. Impressively, almost all n-gram models performed better than any individual handcrafted feature, with the best model, character bi-grams, demonstrating an AUC_{avg} of 75.4. When we inspect the learned logistic regression weights for the best variant of each model type (summarized in Figure 5), we find a few interesting observations:

Top character bi-grams. The highest weighted character bi-grams exclusively aim to identify the beginnings of quotations, agreeing with the success of the quote count feature that the presence of a quote is highly informative. Curiously, the presence of a quotation being present but not starting the sentence is a strong negative indicator (i.e. “ “”).

Bottom character bi-grams. Among the lowest weighted character bi-grams are also indicators of numbers, URLs, and possibly twitter handles (i.e. “@”).

Token	n = 1	n = 2	n = 3
char	70.7	75.4	74.2
word	73.9	72.3	65.6

Table 2: AUC_{avg} scores of the n-gram models.

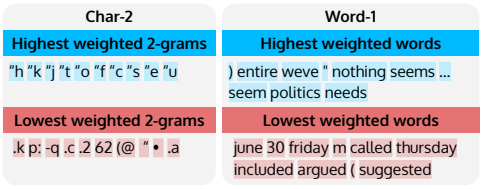


Figure 5: The ten highest and lowest weighted n-grams for the best character and word models.

Words. Although the highest weighted words are difficult to interpret together, among the lowest weighted words are those indicating past tense: "called", "included", "argued", "suggested". This suggests a promising approach for PQ selection includes identification of the tense of each sentence.

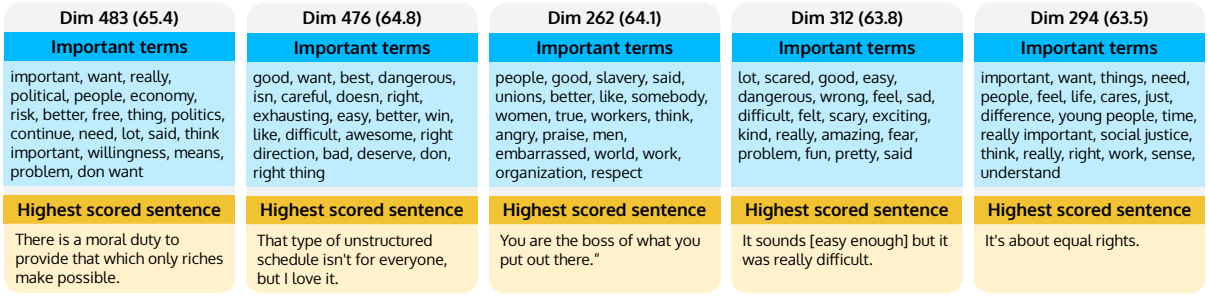
5.3 SBERT Embeddings with a Progression of Neural Architectures

The results of the neural architectures using SBERT embeddings is included in Table 3. Overall, these results suggest that using document embeddings helps performance, especially with a mixture-of-experts architecture. This is seen by the general trend of improved performance from group A to B to C. Within each group, adding the fully connected layers (the "deep" models) helps.

Model	AUC_{avg}	**Width**	**# Params**
A-basic	76.7±0.15	-	7.7E+02
A-deep	77.7±0.16	128, 64	1.1E+05
B-basic	77.1±0.24	-	1.5E+03
B-deep	78.3±0.29	128, 64	2.1E+05
C-basic ($k = 16$)	77.7±0.51	-	2.5E+04
C-deep ($k = 4$)	78.7±0.07	32, 16	5.0E+04

Table 3: Results on the neural architectures. Performance mean and std. dev. is calculated with five trials. k refers to the # experts, only applicable to C group models. Width values correspond to the width of the two additional fully connected layers (only applicable to the deep models).

Inspecting individual SBERT dimensions. Given the performance of these embeddings, we are eager to understand what aspects of the text it picks up on. To do this, we first identify the most informative of the 768 dimensions for PQ selection by training a logistic regression model for each one. For each single-feature model, we group sentences in the test set by PQ probability (high, medium, and low) and perform a TF-IDF analysis to identify key terms associated with *increasing* PQ probability[3]. See Appendix B for more details. Results for the top five best performing dimensions are shown in Figure 6. We find that each of these dimension is sensitive to the presence of a theme (or combination of themes) generally interesting and important to society. Our interpretations of them are: (a) politics and doing the right thing, (b) working hard on difficult/dangerous things, (c) discrimination, (d) strong emotions – both positive and negative, and (e) social justice.



(a) (b) (c) (d) (e)

Figure 6: The top five best performing SBERT embedding dimensions, along with the terms associated with increasing PQ probability with respect to that dimension. For each dimension, we also include the sentence from the test articles which that dimension most strongly scores as being a PQ sentence. At the top of each box is the dimension index and the test AUC_{avg}.

[3]Likewise, we could study terms associated with *decreasing* PQ probability – to deeper understand what *bores* people.

5.4 Cross-Task Models

The results for the cross-task models of headline popularity prediction, clickbait identification, and summarization are shown in Table 4. Considered holistically, the results suggest that PQs are not designed to inform the reader about what they are reading (the shared purpose of headlines and summaries), so much as they are designed to motivate further engagement (the sole purpose of clickbait). However, the considerable performance gap between the clickbait model and PQ-specific models (such as character bi-grams and SBERT embeddings) suggest that this is only one aspect of choosing good pull quotes.

Model	AUC_{avg}
headline popularity	56.9
clickbait	63.8
LexRank	51.9
SumBasic	44.9
KLSum	55.1
TextRank	55.9

Table 4: Performance of the cross-task models.

Another interesting observation is the variability in performance of summarizers at PQ selection. If we consider the summarization performance of these models as reported together in (Chen et al., 2016), we find that PQ selection performance is not strongly correlated with their summarization performance.

5.5 Human Evaluation

As a final experiment, we conduct a qualitative evaluation to find out how well the PQs selected by various models (including the true PQ sources) compare. The results are summarized in Table 5. We randomly select 50 articles from the test set and ask nine volunteers to evaluate the candidate PQs extracted by six different models. They are asked to rate each of the 300 candidate PQs based on how interested it makes them in reading more of the article on a scale of 1 (not at all interested) to 5 (very interested). For each model we report the following metrics: (1) the **rating** averaged across all responses (with 5 being the best), (2) the average **rank** within an article (with 1 being the best), and (3) $\mathbf{1^{st}}$ **Place Pct.** – how often the model produces the best PQ for an article (with 100% being the best).

Model	**Rating** ↑	**Rank** ↓	$\mathbf{1^{st}}$ **Place Pct.** ↑
True PQ Source	2.75	3.04	**28%**
Char-2	**2.86**	**2.74**	**28%**
C-deep	2.75	3.08	18%
Headline pop.	2.57	3.66	8%
Clickbait	2.70	3.26	18%
TextRank	2.69	3.32	14%

Table 5: The results of human evaluation comparing models in terms of how interested the reader is in reading more of the article. The ↑ and ↓ indicate whether better values for a metric are respectively higher or lower.

The results in Table 5 show that the two PQ-specific approaches (Char-2 and C-deep using the best hyperparameters from Section 5.3) perform on par or slightly better than the true PQ sources. By generally out-performing the transfer models, this further supports our claim that the PQ selection task serves a unique purpose. When looking at how often each model scores 1^{st} place, which accentuates their performance differences, we can see that the headline and summarization models in particular perform poorly. Mirroring the results from Section 5.4, among the cross-task models, the clickbait model seems to perform best.

6 Conclusion

In this work we proposed the novel task of automatic pull quote selection as a means to better understand how to engage readers. To lay foundation for the task, we created a PQ dataset and described and benchmarked four groups of approaches: handcrafted features, n-grams, SBERT-based embeddings combined with a progression of neural architectures, and cross-task models. By inspecting results, we encountered multiple curious findings to inspire further research on PQ selection and understanding reader engagement.

There are many interesting avenues for future research with regard to pull quotes. In this work we assume that all true PQs in our dataset are of equal quality, however, it would be valuable to know the quality of individual PQs. It would also be interesting to study how to make a given phrase more PQ-worthy while maintaining the original meaning.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grants Program. NSERC invests annually over $1 billion in people, discovery and innovation.

References

Jennifer M Aquino and Karen M Arnell. 2007. Attention and the processing of emotional words: Dissociating effects of arousal. *Psychonomic Bulletin & Review*, 14(3):430–435. 4

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. " O'Reilly Media, Inc.". 12

Richard Braddock. 1974. The frequency and placement of topic sentences in expository prose. *Research in the Teaching of English*, 8(3):287–302. 4

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904–911. 4, 12

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 9–16. IEEE. 1, 3, 4, 5, 13

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. *arXiv preprint arXiv:1610.08462*. 9

François Chollet et al. 2015. Keras. `https://keras.io`. 12

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 5

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479. 3, 13

Rudolf Flesch. 1979. *How to Write Plain English: A Book for Lawyers and Consumers*. Harper & Row New York, NY. 4

Nigel French. 2018. *InDesign Type: Professional Typography with Adobe InDesign*. Adobe Press. 1

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. 3, 13

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273, Baltimore, Maryland, June. Association for Computational Linguistics. 1

Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360. 3

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. 13

Tim Holmes. 2015. *Subediting and Production for Journalists: Print, Digital & Social*. Routledge. 1

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*. 12

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87. 5

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 12

Johanna Kissler, Cornelia Herbert, Peter Peyk, and Markus Junghofer. 2007. Buzzwords: Early cortical responses to emotional eords during reading. *Psychological Science*, 18(6):475–480. 4

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980. 12

Sotiris Lamprinidis, Daniel Hardt, and Dirk Hovy. 2018. Predicting news headline popularity with syntactic and semantic knowledge using multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 659–664. 1, 2, 4

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 257–266. Association for Computational Linguistics. 3

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165. 3

Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1318–1327. Association for Computational Linguistics. 3

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411. 3, 13

Nuno Moniz and Luís Torgo. 2018. Multi-source social feedback of online news feeds. *arXiv preprint arXiv:1801.07055*. 5, 13

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016a. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*. 3

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016b. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*. 3

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*. 3

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer. 1, 3

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101. 3, 13

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233. 3

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830. 13

Alicja Piotrkowicz, Vania Dimitrova, Jahna Otterbacher, and Katja Markert. 2017. Headlines matter: Using headlines to predict the popularity of news articles on twitter and facebook. In *Eleventh International AAAI Conference on Web and Social Media*. 1, 2, 4

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *European Conference on Information Retrieval*, pages 810–817. Springer. 1, 3, 4

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*. 2, 5, 12

Patrice Lopez Laurent Romary. 2010. Automatic key term extraction from scientific articles in grobid. In *SemEval 2010 Workshop*, page 4. 3

Mark Sadoski, Ernest T Goetz, and Maximo Rodriguez. 2000. Engaging texts: Effects of concreteness on comprehensibility, interest, and recall in four text types. *Journal of Educational Psychology*, 92(1):85. 4, 7

Harald T Schupp, Jessica Stockburger, Maurizio Codispoti, Markus Junghöfer, Almut I Weike, and Alfons O Hamm. 2007. Selective visual attention to emotion. *Journal of Neuroscience*, 27(5):1082–1089. 4

James Glen Stovall. 1997. *Infographics: A Journalist's Guide*. Allyn & Bacon. 1

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40. 3

Peter Turney. 1999. Learning to extract key phrases from text, nrc technical report erb· 1057. Technical report, Canada: National Research Council. 3

Lasya Venneti and Aniket Alam. 2018. How curiosity can be modeled for a clickbait detector. *arXiv preprint arXiv:1806.04212*. 1, 3

Wayne Wanta and Dandan Gao. 1994. Young readers and the newspaper: Information recall and perceived enjoyment, readability, and attractiveness. *Journalism Quarterly*, 71(4):926–936. 1, 7

Wayne Wanta and Jay Remy. 1994. Information recall of four newspaper elements among young readers. 1, 7

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4):1191–1207. 4, 12

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*. 3

Appendix A Implementation Details

Here we outline the various tools, datasets, and other implementation details related to our experiments:

- To perform part-of-speech tagging for feature extraction, we use the NLTK 3.4.5 perceptron tagger (Bird et al., 2009).
- To compute sentiment, the VADER Sentiment Analysis tool is used (Hutto and Gilbert, 2014), accessed through the NLTK library.
- Implementations of the R_{Flesch} readability metric is provided by the Textstat 0.6.0 Python package[4]. The corpus of easy words for $R_{difficult}$ is also made available by this package.
- Valence, arousal word ratings are obtained from the dataset described in (Warriner et al., 2013)[5]. When computing average valence and arousal for a sentence, stop words are removed and when a word rating cannot be found, a value of 5 is used for valence and 4 for arousal (the mean word ratings).
- Concreteness word ratings are obtained from the dataset described in (Brysbaert et al., 2014) [6]. The concreteness score of a sentence is computed similar to valence and arousal, with a mean word rating of 5 used when no value for a word is available.
- For the n-gram models, a vocabulary size of 1000 was used for all models, and lower-casing was applied for the character and word models.
- The SBERT (Reimers and Gurevych, 2019) implementation and pre-trained models are used for text embedding[7].
- All neural networks using the SBERT embeddings were implemented with the Keras library (Chollet and others, 2015) with the Adam optimizer (Kingma and Ba, 2014) (with default Keras settings) and binary cross-entropy loss. Early stopping is done after validation loss stops decreasing for 4 epochs – with a maximum of 100 epochs. In the deep version of the models, we include two additional densely connected layers as shown in Figure 3, with the second additional layer having half the width of the initial one. We use selu activations (Klambauer et al., 2017) for the additional layers

[4]Available online here: `https://github.com/shivam5992/textstat`

[5]Available online at `http://crr.ugent.be/archives/1003`.

[6]Available online at `http://crr.ugent.be/archives/1330`.

[7]Can be found online at `https://github.com/UKPLab/sentence-transformers`. We use the `bert-base-nli-mean-tokens` pre-trained model.

and a dropout rate of 0.5 for only the first additional densely connected layer (Hinton et al., 2012). The hyperparameters requiring tuning for each model and the range of values tested (grid search) is provided in Table A.1.

- The clickbait identification dataset introduced by (Chakraborty et al., 2016) is used, which contains 16,000 clickbait samples and 16,000 non-clickbait headlines[8].
- The headline popularity dataset introduced by (Moniz and Torgo, 2018) is used, which includes feedback metrics for about 100,000 news articles from various social media platforms[9]. For pre-processing, we remove those article where no popularity feedback data is available, and compute popularity by averaging percentiles across platforms. For example, if an article is in the 80^{th} popularity percentile on Facebook and in the 90^{th} percentile on LinkedIn, then it is given a popularity score of 0.85.
- We use the following summarizers: TextRank (Mihalcea and Tarau, 2004), SumBasic (Nenkova and Vanderwende, 2005), LexRank (Erkan and Radev, 2004), and KLSum (Haghighi and Vanderwende, 2009)[10].
- We used the Scikit-learn (Pedregosa et al., 2011) implementations of AdaBoost, decision trees, and logistic regression. To accommodate the imbalanced training data, balanced class weighting was used for the decision trees in Adaboost and logistic regression. For AdaBoost, we use 100 estimators with the default learning rate of 1.0. For logistic regression we use the default settings of L2 penalty with $C = 1.0$.

model	Initial width	# Experts
A-basic	-	-
A-deep	[16, 32, 64, 128, 256, 512]	-
B-basic	-	-
B-deep	[16, 32, 64, 128, 256, 512]	-
C-basic	-	[2, 4, 8, 16]
C-deep	[16, 32, 64, 128, 256, 512]	[2, 4, 8, 16]

Table A.1: Hyperparameter values used in grid search for the different SBERT neural networks. The models with the best performance on the validation set averaged across 5 trials are reported in Table 3.

Appendix B TF-IDF Analysis of SBERT Embedding Dimensions

In order to uncover the key terms associated with increased PQ probability for a given SBERT embedding dimension, the following steps were performed:

1. Train a logistic regression model using that single feature. Make a note of whether the coefficient is positive (i.e. increasing the feature value increase PQ probability) or negative (i.e. decreasing feature value increases PQ probability).
2. Take all test sentences and split them into three groups: (1) those where the feature value is in the top k, (2) those where the feature value is in the middle $2k$, and (3) those where the feature value is in the bottom k. We use $k = 2000$.
3. Join together the sentences within each of the three groups so that we have three "documents" and apply TF-IDF on this set of documents. We use the Scikit-learn (Pedregosa et al., 2011) implementation, with an n-gram range of 1-3 words and use the English stopword list with `sublinear_tf = True`. All other settings are at the default values.

[8] Available online at `https://github.com/bhargaviparanjape/clickbait/tree/master/dataset`.
[9] Available online at `https://archive.ics.uci.edu/ml/machine-learning-databases/00432/Data/`.
[10] Implementations provided by Sumy library, available at `https://pypi.python.org/pypi/sumy`.

4. If the coefficient from step 1 is positive, use the highest ranked terms for group 1. If the coefficient is negative, use the highest ranked terms for group 3.

Appendix C Model-Chosen Pull Quote Examples

Model	Highest rated sentence(s)
True PQ Source	"To date, the fishing industry in British Columbia has not raised the carbon tax as an area of specific concern," it says.
Quote_count	OTTAWA - The federal government's carbon tax could take a toll on Canada's fishing industry, causing its competitiveness to "degrade relative to other nations," according to an analysis from the fisheries department.
Sent_position	In the aquaculture and seafood processing industries, in contrast, fuel makes up just 1.6 per cent and 0.8 per cent of total costs, respectively.
R_difficult	That would result in a difference in the GDP of about $2 billion in 2022, or 0.1 per cent.
POS_PRP	"To date, the fishing industry in British Columbia has not raised the carbon tax as an area of specific concern," it says.
POS_VB	"The relatively rapid introduction of measures to reduce GHG emissions would allow little time for industry and consumers to adjust their behaviour, creating a substantial risk of economic disruption and uncertainty."
A_concreteness	"This could have a negative impact on the competitiveness of Canada's fishing industry."
Char-2	"However, Canada's competitiveness may degrade relative to other nations that have not yet announced plans, or are proceeding more slowly towards measures to reduce GHG emissions," the memo says.
Word-1	The memo concludes that short-term impacts are expected to be "low to moderate," and the department will "continue to monitor developments."
C-deep	"To date, the fishing industry in British Columbia has not raised the carbon tax as an area of specific concern," it says.
Headline popularity	The four largest provinces - Quebec, Ontario, Alberta and B.C.
Clickbait	Ottawa has said all jurisdictions that don't have their own carbon pricing plans in place this year will have the federal carbon tax imposed on them in January 2019, starting at 20$pertonneandincreasingto$50 per tonne in 2022.
TextRank	The analysis was completed in December 2016, shortly after most provinces and territories had signed Ottawa's pan-Canadian climate change framework, committing them to a range of measures, including carbon pricing, to reduce Canada's 2030 emissions to 30 per cent below 2005 levels.

Table C.1: Article source: `https://nationalpost.com/news/politics/federal-carbon-tax-could-degrade-canadian-fishing-industrys-competitiveness-says-memo`.

Model	Highest rated sentence(s)
True PQ Source	I think so many people voted for me because I think they're just proud of me as well.
Quote_count	The school year is finally coming to an end and that means it's prom season, woo season!
Sent_position	I texted my friends like, "Oh my god I'm freaking out.
R_difficult	I'm only at the school for an hour and a half every other day so I had no idea that we were even voting.
POS_PRP	I think so many people voted for me because I think they're just proud of me as well.
POS_VB	- and some people would send me them, but I just choose not to read them.
A_concreteness	I didn't hear about anything.
Char-2	Something that I just want everyone to take away from this is you can be you as long as you're not hurting anyone else and as long as you're not breaking any rules.
Word-1	Something that I just want everyone to take away from this is you can be you as long as you're not hurting anyone else and as long as you're not breaking any rules.
C-deep	I don't think there's any day where I haven't worn a full face of makeup to school, and I always dress up.
Headline popularity	I think so many people voted for me because I think they're just proud of me as well.
Clickbait	I texted my friends like, "Oh my god I'm freaking out.
TextRank	In an interview with Cosmopolitan.com, he talked about putting together his look, why he didn't see his crowning coming, and what he'd like to tell the haters.

Table C.2: Article source: `https://www.cosmopolitan.com/lifestyle/a20107039/south-carolina-prom-king-adam-bell-interview/`

Model	Highest rated sentence(s)
True PQ Source	There is not a downtown in the whole wide world that's made better by vehicle traffic.
Quote_count	We need to stop widening roads and otherwise "improving" our road infrastructure, and pronto.
Sent_position	By putting an immediate moratorium on it.
R_difficult	But at the same time (this is the important part), make it super easy, free (or nearly free) and convenient to get around downtown.
POS_PRP	Not, I think, if we have any say over it.
POS_VB	Have them criss-cross the inner core.
A_concreteness	Not, I think, if we have any say over it.
Char-2	We live far away from where we need to be, and we enjoy activities that aren't always practical by bus, especially if you happen to have kids that need to be in six different places every day.
Word-1	We live far away from where we need to be, and we enjoy activities that aren't always practical by bus, especially if you happen to have kids that need to be in six different places every day.
C-deep	I want to scream.
Headline popularity	Personally, I'd rip out the Queensway and turn it into a light-rail line with huge bike paths, paths for motorcycles, and maybe a lane or two dedicated to autonomous vehicles and taxis and ride-shares.
Clickbait	It's an idea I've been obsessed with since visiting Portland, Oregon, in 2004.
TextRank	Not, I think, if we have any say over it.

Table C.3: Article source: `https://ottawacitizen.com/opinion/columnists/armchair-mayor-fewer-cars-more-transit-options-would-invigorate-ottawa`

Model	Highest rated sentence(s)
True PQ Source	But Pelosi seems to have thought more about alliteration than what pitch would effectively challenge the inaccurate but narratively satisfying story the president had just told.
	Sanders packed more visceral humanity in the first minute or so of his remarks than in the entirety of Pelosi and Schumer's response.
	And perhaps most importantly, he validated that there is, in fact, a crisis afoot: one created by Trump, as well as several produced by structural forces the political class has long ignored.
	And this is an important point: The temptation to fact-check is understandable. And a certain amount of fact-checking is necessary to keep Trump accountable. But poking holes in Trump's narrative, by itself, is not enough.
Quote_count	The life of an American hero was stolen by someone who had no right to be in our country," he said.
Sent_position	An opioid crisis does kill thousands of Americans each year.
R_difficult	The life of an American hero was stolen by someone who had no right to be in our country," he said.
POS_PRP	I'm not going to blame you [Chuck Schumer] for it."
POS_VB	I live paycheck to paycheck, and I can't get a side job because I still have to go to my unpaid federal job."
A_concreteness	He didn't disappoint.
Char-2	"Let me be as clear as I can be," said Sanders, "this shutdown should never have happened."
Word-1	"Let me be as clear as I can be," said Sanders, "this shutdown should never have happened."
C-deep	All are equally guilty - children are merely "pawns," not people.
Headline popularity	And what Trump said about who is hurting most is true: "Among the hardest hit are African-Americans and Hispanic-Americans."
Clickbait	"[Trump] talked about what happened the day after Christmas?
TextRank	These are people in the FBI, in the TSA, in the State Department, in the Treasury Department, and other agencies who have, in some cases, worked for the government for years."

Table C.4: Article source: `https://theintercept.com/2019/01/09/trump-speech-democratic-response/`. This article demonstrates a case where there are many real PQs in an article. It also highlights the need for future work which can create multi-sentence PQs (True PQ #4 consists of two sentences).