News Article Teaser Tweets and How to Generate Them

Sanjeev Kumar Karn[1,2], **Mark Buckley**[2], **Ulli Waltinger**[2] **and Hinrich Schütze**[1]

[1]Center for Information and Language Processing (CIS), LMU Munich
[2]Machine Intelligence, Siemens CT, Munich, Germany
[1]skarn@cis.lmu.de
[2]{sanjeev.kumar_karn,mark.buckley,ulli.waltinger}@siemens.com

Abstract

In this work, we define the task of teaser generation and provide an evaluation benchmark and baseline systems for the process of generating teasers. A teaser is a short reading suggestion for an article that is illustrative and includes curiosity-arousing elements to entice potential readers to read particular news items. Teasers are one of the main vehicles for transmitting news to social media users. We compile a novel dataset of teasers by systematically accumulating tweets and selecting those that conform to the teaser definition. We have compared a number of neural abstractive architectures on the task of teaser generation and the overall best performing system is See et al. (2017)'s seq2seq with pointer network.

1 Introduction

A considerable number of people get their news in some digital format.[1] The trend has made many publishers and editors shift their focus to the web and experiment with new techniques to lure an Internet-savvy generation of readers to read their news stories. Therefore, there has been a noticeable increase in the sharing of short illustrative pieces of texts about the news on social media.

We define a ShortcutText as a short text (about 15 words or less) describing and pointing to a news article and whose purpose is to invite the recipient to read the article. A headline is a ShortcutText that optimizes the relevance of the story to its reader by including interesting and high news value content from the article (Dor, 2003). Click-bait is a pejorative term for web content whose main goal is to make a user click an adjoining link by exploiting the information gap. According to the definition, a principal part of the headline is an extract of the article, thereby creating an impression of the upcoming story. However, click-bait, a ShortcutText, contains mostly elements that create anticipation, thereby making a reader click on the link; however, the reader comes to regret their decision when the story does not match the click-bait's impression (Blom and Hansen, 2015). Thus, click-bait provides a false impression (non-bona fide) and contains insufficient information (highly abstractive).

	bona-fide	teasing	abstractive
headline	yes	no	low
clickbait	no	yes	high
teaser	yes	yes	high

Table 1: The table shows three categories of ShortcutTexts and their properties

We introduce the new concept of *teaser* and define it as a ShortcutText devised by fusing curiosity-arousing elements with interesting facts from the article in a manner that concurrently creates a valid impression of an upcoming story and a sense of incompleteness, which motivates the audience to read the article. A teaser is one of the main vehicles for transmitting news on social media. Table 2 shows some teasers from a popular newswire *The Wall Street Journal*.

We also introduce properties such as teasing, abstractive, and bona-fide, which not only differentiate teasers from other ShortcutTexts but also help in compiling a dataset for the study. Teasing indicates whether curiosity-arousing elements are included in the ShortcutText. Abstractive indicates whether a fair proportion of the ShortcutText is distilled out of the news article. Bona-fide answers whether the news story matches the impression created by the ShortcutText. Table 1 lists the common forms of the ShortcutTexts along with the presence or absence of the properties mentioned

[1] http://www.journalism.org/2008/07/21/the-influence-of-the-web/

Proceedings of NAACL-HLT 2019, pages 3967–3977
Minneapolis, Minnesota, June 2 - June 7, 2019.

Article	Global trade is in trouble, and investors dont seem to care. One of the ironies of the election of a fierce nationalist in the U.S. . . .
Headline	Steel Yourself for Trumps Anti-Trade Moves
Teaser	Investors don't seem worried about a trade war. Could tariffs by Trump start one?
Article	The U.S. Supreme Court on Monday partially revived President Donald Trumps executive order suspending travel from six countries . . .
Headline	High Court Says Travel Ban Not For Those With 'Bona Fide' Relationships
Teaser	In a 'bona fide' relationship? You can visit the U.S.
Article	Gan Liping pumped her bike across a busy street, racing to beat a crossing light before it turned red. She didnt make it. . . .
Headline	China's All-Seeing Surveillance State Is Reading Its Citizens' Faces
Teaser	China is monitoring its citizens very closely. Just ask jaywalkers.

Table 2: The table contains tuples of news articles and their ShortcutTexts: headline and teaser. These tuples are from a popular newswire, *The Wall Street Journal*.

above.

In this study, we focus on teasers shared on Twitter[2], a social media platform whose role as a news conduit is rapidly increasing. An indicative tweet is a Twitter post containing a link to an external web page that is primarily composed of text. The presence of the URL in an indicative tweet signals that it functions to help users decide whether to read the article, and the short length confirms it as a ShortcutText like a headline or teaser. Lloret and Palomar (2013) made an early attempt at generating indicative tweets using off-the-shelf extractive summarization models, and graded the generated texts as informative but uninteresting. Additionally, Sidhaye and Cheung (2015)'s analysis showed extractive summarization as an inappropriate method for generating such tweets as the overlaps between the tweets and the corresponding articles often are low. Our study shows that teasers, bona fide indicative tweets, do exhibit significant, though not complete, overlaps, and, therefore, are not appropriate for extractive but certainly for abstractive summarization.

Our contributions:

1) To the best of our knowledge, this is the first attempt to compare different types of ShortcutTexts associated with a news article. Furthermore, we introduce a novel concept of a teaser, an amalgamation of article content and curiosity-arousing elements, used for broadcasting news on social media by a news publisher.

2) We compiled a novel dataset to address the task of teaser generation. The dataset is a collection of news articles, ShortcutTexts (both teasers and headlines), and story-highlights. Unlike ShortcutText, a story-highlight is brief and includes self-contained sentences (about 25-40 words) that allow the recipient to gather information on news stories quickly. As all corpora based on news articles include only one of these short texts, our dataset provides the NLP community with a unique opportunity for a joint study of the generation of many short texts.

3) We propose techniques like unigram overlap and domain relevance score to establish abstractivity and teasingness in the teasers. We also apply these techniques to headlines and compare the results with teasers. The comparison shows teasers are more abstractive than headlines.

4) High abstractivity makes teaser generation a tougher task; however, we show seq2seq methods trained on such a corpus are quite effective. A comparison of different seq2seq methods for teaser generation shows a seq2seq combining two levels of vocabularies, source and corpus, is better than one using only the corpus level. Therefore, we set a strong baseline on the teaser generation task with a seq2seq model of See et al. (2017).

The remaining paper is structured as follows. In Section 2, we provide a detailed description of the data collection and analyses. In Section 3, we describe and discuss the experiments. In Section 4, we describe a user study of model-generated teasers. In Section 5, we discuss the related works. Section 6 concludes the study.

2 Teaser Dataset

Several linguistic patterns invoke curiosity, e.g., provocative questions and extremes for comparison. A retrieval of teasers from a social media platform using such patterns requires the formulation of a large number of complex rules as these patterns often involve many marker words and correspondingly many grammar rules. A computationally easy approach is to compile circulations from bona-fide agents involved in luring business on such media, and then filtering out those that don't comply with defined characteristics of a teaser; see Table 1. We followed the latter approach and chose Twitter to conduct our study.

2.1 Collection

We identified the official Twitter accounts of English-language news publications that had

[2] https://twitter.com/

tweeted a substantial number of times before the collection began; this removes a potential source of noise, namely indicative tweets by third-party accounts referencing the articles via their URL. See supplementary A.1 for the list of Twitter accounts. We downloaded each new tweet from the accounts via Twitter's live streaming API. We limited the collection to indicative tweets and extracted the article text and associated metadata from the webpage using a general-purpose HTML parser for news websites.[3] Overall, we collected approximately 1.4 million data items.

2.2 Analysis

We propose methods that evaluate teasingness and abstractivity in the teasers and verify them through analyses. We then combine those methods and devise a teaser recognition algorithm. Analyses are performed on lowercase, and stopwords-pruned texts.

2.2.1 Extractivity

For a given pair of strings, one is an extract of another if it is a substring of it. Teasers are abstractive, which we confirm by making sure that the ShortcutText is not an extract of article sentences. Additionally, a teaser of an article is designed differently than the headline; therefore, they must be independent of one other, i.e., non-extractive.

2.2.2 Abstractivity

Abstractivity, a principle characteristic of the teaser, implies that the teaser should exhibit content overlap with its source, but not a full overlap.

We rely on Sidhaye and Cheung (2015)'s method of computing the percentage match between two stemmed texts for grading abstractivity. We obtain unigrams of the first, X_1, and second text, X_2, using function $uni(X)$ and compute the percentage match using Eq. 1:

$$perc\_match(X_1, X_2) = \frac{|uni(X_1) \cap uni(X_2)|}{|uni(X_1)|} \tag{1}$$

Given a ShortcutText and article, initially, a sequence of texts is obtained by sliding a window of size p on the article sentences. Then, $perc\_match$ scores between the ShortcutText and sequence of texts are computed. A text with the highest score is selected as the prominent section for the ShortcutText in the article.

Article	Diabetes medication, such as insulin, lowers blood sugar levels and . . .. But experts have revealed a natural treatment for diabetes could be lurking in the garden. . . . Fig leaves have properties that can help diabetes . . . . An additional remedy . . .
headline	Diabetes treatment: Natural remedy made from fig leaves revealed
Teaser	**Would** you **Adam** and **Eve** it? Natural treatment for DIABETES could be **growing** in your garden

Table 3: ShortcutTexts and their non-overlaps (bold).

A full-overlap, i.e., $perc\_match$ of 1 is likely to be a case where the ShortcutText disseminates information of its prominent section. However, a non-overlap is very likely to be click-bait or noise. Thus, we filter out instances where the match score between a ShortcutText, potential teaser, and its prominent section is above 80% or below 20%. The intuition for the filtering is that the teasing words are likely to be absent from the prominent section, and an absence of a minimum of 2-3 words (often 20%) is the easiest way to ascertain this fact. Table 3 shows an example. Analogously, a presence of a minimum of 2-3 words from the source asserts that it is not click-bait or noise.

We use the sliding window size, p, of 5,[4] and filter the data instances where the $perc\_match$ between the tweet and prominent section is lower than 0.2 or greater than 0.8.

2.2.3 Teasingness

Apart from abstractivity, teasers include words and phrases that tease and are are embedded by authors who often draw on their vast knowledge of style and vocabulary to devise teasers. A commonly recognizable pattern among them is the inclusion of unusual and interesting words in a given context, e.g., words like *Adam* and *Eve* in the example of Table 3.

The Pareto principle or the law of the vital few, states that the 2,000 of the most frequently used words in a domain cover about 80% of the usual conversation texts (Nation, 2001; Newman, 2005). At first glance, filtering those abstractive ShortcutTexts that constitute only frequent words should intuitively prune uninteresting ones and save ones that are similar to the example in Table 3. However, a closer look at the pruned ShortcutTexts shows several interesting teasers with substrings comprised of out-of-place frequent-words, e.g., *Las Vegas gunman Stephen bought nearly %%*

[3] https://github.com/codelucas/newspaper/

[4] Most of the prominent information is supposedly within a few leading sentences in the news articles due to the inverted pyramid news writing style.

guns legally. But none of the purchases set off any red flags, with an interesting sentence fragment containing the phrase *red flags*. This suggests that the methodology that uses plain frequency of words is not sufficient for determining interesting information.

$$\begin{aligned} tf_{domain}(w,d) &= \frac{|\text{term } w \text{ in domain } d|}{|\text{terms in domain } d|} \\ idf_{domain}(w) &= log\, \frac{|\text{domains}|}{|\text{domains containing } w|} \\ dr(w,d) &= tf_{domain}(w,d) \times idf_{domain}(w) \end{aligned} \tag{2}$$

Thus, we look at unusualness at a level lower than the corpus. We rely on domain relevance (dr) (Schulder and Hovy, 2014), an adapted TF-IDF (term frequency inverse document frequency) metric that measures the impact of a word in a domain and, therefore, identifies unusual words in a specific domain, and is computed using Eq. 2.

A word is assigned a very low dr score if the word is either non-frequent in the domain and too frequent among other domains (unusualness) or non-frequent in all domains (rare); see Table 4. As a very low dr score corresponds to unusualness, a presence of very low dr values among the non-overlapping words of the ShortcutText suggest a high likelihood of it being a teaser, and therefore, we compile them as teasers. However, the filtering requires a threshold dr value that defines anything lower than it as a very low dr. Also, computing dr requires domain information of the text.

	$freq$_Out	$\neg$ $freq$_Out
$freq$_IN	low	high
$\neg$ $freq$_IN	very low	very low

Table 4: The table shows dr score range. IN and Out refer to in-domain and out-of-domain respectively.

Would	**Adam**	**Eve**	Natural	treatment
0.104	**0.0027**	**0.0025**	0.025	0.016
DIABETES	could	**growing**	garden	
0.005	0.105	0.022	0.01	

Table 5: Teaser words and their dr values. Non-overlaps are in bold blue.

Obtaining Domains

We make use of articles and their keywords to determine domains. Keywords are meta-information available for a subset of corpus instances. We rely on Doc2vec (Le and Mikolov, 2014) for obtaining the representations for the articles and cluster these representations by K-Means clustering (Hartigan and Wong, 1979).

We rely on elbow criterion and uniformity among keywords in the clusters to determine the number of clusters. The uniformity is validated by manual inspection of 100 most-frequent keywords. Clustering the corpus into eight domains resulted in the final abrupt decrease of the Sum of Squared Error (SSE) as well as uniformly distributed keyword sets. See Table 6 for domain-wise keywords and other statistics.

Selecting a Threshold

We use the domain information and compute dr values of potential teaser texts in the corpus. Table 5 shows nonstop words and dr scores for Table 3 example. Evidently, unusual words have very low dr scores (bold values).

To determine an appropriate threshold, we design an unsupervised methodology based on the Pareto principle. The cue remains the same, i.e., a right threshold will filter only the teasers, and the non-overlapping words in them are less likely to be frequent words.

Thus, we define a range of possible threshold values, and for each value, we compile a corpus of teasers where a non-overlapping word has dr below it. Meanwhile, we also compile sets of most-frequent words that cover 80% of the total word occurrences in all 8 domains (sizes $\approx$ 2000). Then, we determine the ratio of the teasers that have their non-overlapping words completely overlapping the frequent word sets. Finally, we select a value which has the least overlap as the threshold; see Figure 1. We chose 0.005 as it is the boundary below which there is no overlap. We apply this value to abstractive ShortcutTexts and obtain a teaser corpus.

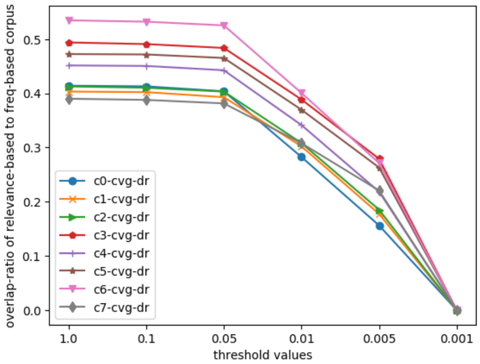


Figure 1: Overlap-ratio of frequency-based and threshold-based filtered teasers for domains (c#).

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Keywords	politics, UK, world, Brexit, Europe, UK Politics, Theresa May	Trump, United States, election, immigration, White House	culture, travel, home and garden, food and beverage	entertainment, celebrities, movies, concerts, Netflix	Europe, Russia, North Korea, Diplomacy, Conflicts	shooting, police, murder, killed, dead, fire, suspect, crash	Corporate, business, Company, automotive, Equities	Sport, Football, Premier League, NFL, Dallas Cowboys, Rugby
Avg. size (words)	763	842	526	838	886	651	791	741

Table 6: The table shows clusters of domains and corresponding frequent-keywords and average article size (words) in them.

2.3 Teaser Recognition Algorithm

We combine the above three methodologies and devise a teaser recognition algorithm; see Algorithm. 1.

We use notations like uppercase bold for a matrix, lowercase italic for a variable and uppercase italic for an array. A data instance in the corpus has an article A, headline H, tweet T, and domain d. An article, A, has a sequence of sentences, $S = \langle S_1, \ldots, S_{|A|} \rangle$, and each sentence, S_i, has a sequence of words, $\langle w_1, \ldots, w_{|S_i|} \rangle$. WINDOW takes a sequence of sentences, S, and returns a sequence of texts, Z, of size $\frac{|S|-p}{q} + 1$, where p and q are window size and sliding step respectively. The domain-wise dr values for words in the vocabulary, U, is stacked into a matrix, $\mathbf{D}$. IS_TEASER takes $\mathbf{D}$ and items of a data instance, and determines whether its tweet, T, is a teaser.

Algorithm 1 Teaser Recognition

```
1:  procedure IS_TEASER($A, H, T, d, \mathbf{D}$)
2:      if ($T$ in $H$) Or ($H$ in $T$) then
3:          return False                              ▷ sub-string match
4:      for $S$ in $A$ do
5:          if ($T$ in $S$) Or ($S$ in $T$) then
6:              return False                          ▷ sub-string match
7:      $Z \leftarrow$ WINDOW($A, p = 5, q = 1$)
8:      $V \leftarrow$ Array()
9:      for $i$ = 1 to $|Z|$ do
10:         $V$.ADD($perc\_match(Z[i], T)$)           ▷ see Eq. 1
11:     if $max(V) > 0.8$ Or $max(V) < 0.2$ then:
12:         return False                              ▷ abstractivity
13:     $\hat{Y} \leftarrow Z[max(V)]$                ▷ prominent section
14:     $T' \leftarrow \hat{Y} \setminus T$           ▷ non-overlap
15:     $L \leftarrow \mathbf{D}[d;T']$               ▷ indexing
16:     if any ($L < 0.005$) then
17:         return True                               ▷ teasingness
18:     return False
```

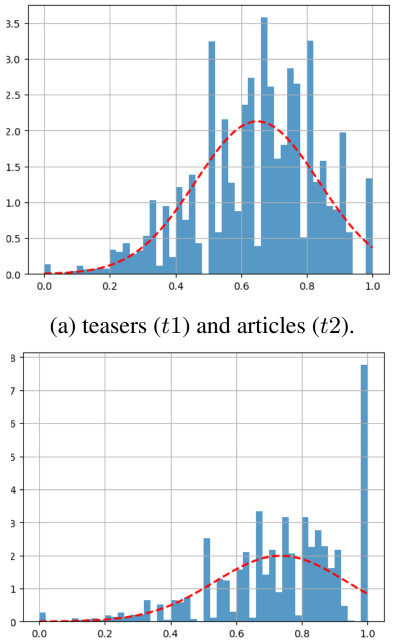
Overall, in Algorithm. 1, steps 2 to 6 checks Extractivity, steps 7 to 12 checks Abstractivity, and steps 13 to 17 checks Teasingness. Table 7 shows the percentage distribution of the total data points that are pruned by each of those analyses. Finally, we compile the remaining 23% data points, i.e., 330k as a teaser corpus.

Analysis		% pruned
Extractivity	wrt headline	37%
	wrt article	5%
Abstractivity		22%
Teasingness		13%

Table 7: The table shows different analyses performed in Algorithm. 1 and the corresponding approximate percentage of data points pruned using them. "wrt" = with respect to.

2.4 Comparing ShortcutTexts

The two ShortcutTexts, headline and teaser, have distinct conveyance mediums and therefore are designed differently, e.g., mean lengths of 10 and 14 respectively. However, abstractivity is also presumed for the headline. Therefore, we conduct additional overlap-based studies to understand the



(a) teasers ($t1$) and articles ($t2$).

(b) headlines ($t1$) and articles ($t2$)

Figure 2: Histogram of unigram overlaps obtained using Eq. 1. The histograms are normalized, i.e., the area under the curve ($\sum$bin-height×bin-width) sum up to one.

differences in the abstractive property between them. We compute and plot the distribution of the overlaps between teasers (T_1) and articles (T_2), and one between headlines (T_1) and articles (T_2); see Figure 2a and Figure 2b for respective plot. Clearly, compared to the teaser, headline distribution is left-skewed (mean 74% and std 20%), and thereby implies that headlines have a lesser abstractive value than teasers.

Further, a review of a few instances of headline-article instances with lesser than 60% overlap reveals cases of noisy headlines or HTML-parse failures; therefore, in a typical scenario a headline with a size of 10 words takes nearly all of its content (≈80%) from the source while a teaser of size 14 has sufficient non-extractive contents (≈32%). See Table 3 for an example.

3 Experiments

3.1 Models

We experiment with two state-of-the-art neural abstractive summarization techniques, attentive seq2seq (Bahdanau et al., 2014) and pointer seq2seq (See et al., 2017), for teaser generation. Attentive seq2seq learns to generate a target with words from a fixed vocabulary, while pointer seq2seq uses a flexible vocabulary, which is augmented with words from the source delivered through the pointer network. We refer to the individual papers for further details.

Evaluation Metrics: Studies on text-summarization evaluate their system using Rouge; therefore, we report Rouge-1 (unigram), Rouge-2 (bigram), and Rouge-L (longest-common substring) as the quantitative evaluation of models on our corpus.

Parameters: We initialized all weights, including word embeddings, with a random uniform distribution with mean 0 and standard deviation 0.1. The embedding vectors are of dimension 100. All hidden states of encoder and decoder in the seq2seq models are set to dimension 200. We pad short sequences with a special symbol $\langle PAD \rangle$. We use Adam with initial learning rate .0007 and batch size 32 for training. Texts are lowercased and numbers are replaced by the special symbol %. The token length for the source is limited to 100 and target sequence to 25. The teaser baseline experiments and headline generation use vocabulary size of 20000.

	Rouge-1	Rouge-2	Rouge-L
ABS	29.55	11.32	26.42
ABS+	29.76	11.88	26.96
RAS-Elman	33.78	**15.97**	31.15
Nallapati et al.	32.67	15.59	30.64
seq2seq	31.21	12.96	28.87
seq2seq_point	**34.81**	15.59	**32.05**

Table 8: Rouge scores on the standard task of Headline Generation (Gigaword). seq2seq and seq2seq_point are reimplementations of Bahdanau et al. (2014) and See et al. (2017) respectively.

3.2 Baseline Setting

As we reimplemented (Bahdanau et al., 2014) and (See et al., 2017) models, we initially evaluate them on a standard task of headline generation.[5] We use popular headline generation corpus, Gigaword (Napoles et al., 2012), with 3.8M training examples. We fetched the test set from Rush et al. (2015) and report the results on it. The results are compared with the state-of-the-art headline generation methods like Nallapati et al. (Nallapati et al., 2016), ABS (Rush et al., 2015), ABS+ (Rush et al., 2015), and RAS-Elman (Chopra et al., 2016). Since our aim for this experiment is to demonstrate the strength of the models, we limit the model parameters to the extent that we produce comparable results in less computation time. Table 8 compares performances of seq2seq and seq2seq_pointer models with other state-of-the-art methods. The results indicate that the implementations have performance competitive with other state-of-the-art methods.

	Validation		
	Rouge-1	Rouge-2	Rouge-L
seq2seq	15.77	03.52	13.53
seq2seq_point	**21.57**	**07.03**	**18.64**
		Test	
seq2seq	15.26	03.38	13.15
seq2seq_point	**21.05**	**07.11**	**18.49**

Table 9: Rouge F1 scores for seq2seq model and seq2seq_point models on the teaser task.

These models are then trained and evaluated on the teaser corpus obtained using Algorithm. 1 that initially has 330k instances. We then sample 255k instances that have all associated short texts in them. The sampled corpus is split into

[5] codes for collection, analyses and experiments: https://github.com/sanjeevkrn/teaser_collect.git and https://github.com/sanjeevkrn/teaser_generate.git

non-overlapping 250k, 2k and 2k sets for training, validation, and testing, respectively. The split is constructed such that training, validation and test sets have equal representation of all eight domains. Any instances that describe events that were also described in training are removed from validation and test sets; thus, instances encountered in validation / test are quite distinct from instances encountered in training. Models were selected based on their performance on the validation set. Table 9 shows the performance comparison. Clearly, seq2seq_point performs better than seq2seq due to the boost in the recall gained by copying source words through the pointer network.

	Teaser		
	Rouge-1	Rouge-2	Rouge-L
seq2seq	15.26	03.38	13.15
seq2seq_point	**21.05**	**07.11**	**18.49**
	Headline		
seq2seq	18.52	05.34	16.74
seq2seq_point	**23.83**	**08.73**	**21.68**
	Highlights		
seq2seq	31.18	17.57	27.30
seq2seq_point	**35.92**	**22.44**	**31.53**

Table 10: Rouge F1 scores for seq2seq model and seq2seq_point models on the teaser, headline and highlights generation task.

Additionally, models are also trained and evaluated on the other short texts that are available in the novel corpus: headlines (also a ShortcutText) and story-highlights. All the model parameters remain the same except the generation size, which depends on the short text average size, e.g., 35 for highlights. Table 10 compares the performance on the test data. Clearly, seq2seq_point performs better than seq2seq for all the types of short texts. Additionally, the change in the rouge scores with the change of dataset, i.e., Teaser<Headline<Highlights, also corresponds to the level of distillation of source information in them.

Table 11 shows an example of a data instance in the corpus and seq2seq_point model generations. Among generations, only headline and teaser have non-overlapping words; however, the headline non-overlap, *says*, is a frequent word with a high dr (0.11) while the teaser non-overlap, *catch*, is a domain-wise non-frequent one, and therefore, has a very low dr (0.006).

Article	Millions of disease carrying mosquitoes are to be freed in a well-meaning bid . . .. The lab-grown versions are infected with a disease which prevents natural mosquito . . . . But some activists fear the disease could transfer to humans ultimately making all human males sterile . . . . Despite claims it is safe for humans , there are also some concerns . . . rendering humans unable to breed . . .
	Ground-Truth
Headline	**SHOCK** CLAIM: Lab created super-mosquitos released into wild could 'make all men infertile'
Highlight	A **NEW** lab-designed mosquito being released into the wild could **end** the human race by making men sterile, it was claimed **today**.
Teaser	**PLAYING GOD**? Millions of lab-grown diseased mosquitoes to be released into **wild**
	Generated
Headline	millions of mosquitoes could be freed , study **says**
Highlight	millions of disease carrying mosquitoes are to be freed in a well-meaning bid to decimate natural populations of the malaria-spreading insects .
Teaser	activists fear the disease could be freed - but there 's a **catch**

Table 11: seq2seq_pointer generated examples. Non-overlapping words are in bold blue. More examples in supplementary A.2.

Further, the teaser is the most detached from the core news information among the three generations, while still being relevant. The generated highlight is extractive, and this is a reason for relatively high Rouge scores for highlights (see Table 10). Rouge is an overlap-based measure and, therefore, is inclined towards extractive datasets.

3.3 Impact of very low dr

We performed additional experiments to study the impact that can be generated using the domain relevance (dr). All the settings are kept intact as in Section 3.2 except the training corpus; this is changed by increasing the proportion of very low dr (<0.005) terms in the teasers. New models are trained using equal size training instances sampled out of the revised corpora.

A bucketing of very low dr percentages into [0%, 25%), [25%, 35%), [35%, 45%), [45%, 55%) and [55%, 100%) divides the corpus into approximately equal sizes. Also, the mean and standard deviation of teaser-article overlap ratio is nearly equal in all the buckets, i.e., 0.559±0.148, 0.559±0.146, 0.564±0.146, 0.566±0.142, 0.566±0.146, respectively. Thus, the range of buckets corresponds to a range in the percentage of uncommon words. We evaluate the precision and recall of the models. Recall ($|overlap|/|ground\text{-}truth|$) estimates the model capacity in recovering the ground-truth content, while precision ($|overlap|/|generation|$) estimates the relevancy in the generation.

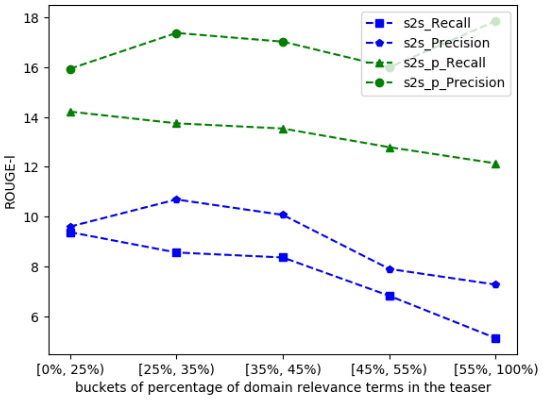


Figure 3: Variation in ROUGE-l on increasing the proportion of domain-relevant terms in the teasers. Models trained on 40k sampled instances.

• pres . trump lashed out on twitter at the hosts of " msnbcs morning "
• migration agency says more than %% people drowned and presumed dead in myanmar to bangladesh
• computer glitch led to google to be dramatically undervalued this morning
• alt-right activist jason kessler says he was swarmed by a group of charlottesville
• of identical triplets who beat the incredible odds of %%% million to survive
• singer and guitar player who declined to appear on britain 's got talent

Table 12: The table shows seq2seq_point generated teasers used in the survey-based study. More examples in supplementary A.2.

	On Twitter		Stimulating	
	Mean	Std	Mean	Std
ground-truth	0.660	0.064	0.621	0.079
seq2seq_point teaser	0.588	0.078	0.559	0.089
baseline	0.476	0.127	0.501	0.111

Table 13: The table shows the mean values and standard deviations of the likelihood of being social-media text and stimulating for users to read. Baseline = lead sentences

As shown in Figure 3, the test recall for both models decreases with the increase in uncommon words in their training. An increase in the proportion of uncommon words makes the models also generate uncommon words, which are not likely to match the ground-truth, thereby reducing the recall. However, in extreme cases, i.e., [45%, 100%), not only training teasers get slightly shorter but also a relatively large proportion of out-of-vocabulary (UNK) is introduced in them, and thereby in the generations. The UNK appears for novel informative words, which are rare words with a very low *dr* as well (see Table 4). Unlike seq2seq, seq2seq_pointer recovers those from the source using pointer network and thus doesn't suffer an abrupt drop in the scores.

Further, the precision scores in extreme cases have a slightly different trend than recall scores, and this is due to shorter generations, which supports precision, but is irrelevant for recall.

4 Human Evaluation

The quantitative evaluations show that state-of-the-art models perform moderately on the novel task. This is mostly due to deficiencies of Rouge, which fails to reward heterogeneous contents. We took a closer look at some of the generated examples, see Table 12, and observed frequent cases where the generation suffered from the typical seq2seq issues, e.g., repetition of words; however, there are also cases where generation is more distinctive than ground-truth and is well formed too. Thus, we carried out a small user study to understand the quality of the generated teasers; however, we only selected non-repeating and non-UNK generations to anonymize the source. The participants in the user study are undergraduate or graduate students with some computer science background and familiarity with social media platforms. Additionally, all the participants have used or have been using twitter.

We assembled a set of texts by randomly sampling 40 seq2seq_point teasers, 40 ground-truth teasers, and 40 lead sentences (baseline), and also established equal representation of the domains. We then assigned 72 sentences (3 per domain per category) to ten participants and asked them to rate texts for two questions: 1) How likely is it that the text is shared on Twitter for a news story by a news organization? and 2) How likely is it that the text makes a reader want to read the story? The first question helps us recognize the participant's understanding of the teasers, as an informed reader will rate a ground-truth significantly higher than the baseline, and 8 of them recognized it correctly, and their ratings are selected for the evaluation. The second question provides a cue as to the model capacity in generating teasing texts by learning interesting aspects present in the teaser corpus.

The annotators rated samples on a scale of 1 to

5; however, we normalized the ratings to avoid the influence of annotators having different rating personalities. The results, summarized in Table 13, show that the human written teasers are most likely to be recognized as social media texts due to their style, which is distinct from the lead sentence; the model trained on such teasers closely follows it. Similarly, human written teasers are good at stimulating readers to read a story compared to the lead sentence and the generated teasers.

5 Related Work

There are two kinds of summarization: abstractive and extractive. In abstractive summarization, the model utilizes a corpus-level vocabulary and generates novel sentences as the summary, while extractive models extract or rearrange the source words as the summary. Abstractive models based on neural sequence-to-sequence (seq2seq) (Rush et al., 2015) proved to generate summaries with higher Rouge scores than the feature-based abstractive models. The integration of attention into seq2seq (Bahdanau et al., 2014) led to further advancement of abstractive summarization (Nallapati et al., 2016; Chopra et al., 2016; See et al., 2017).

There are studies utilizing cross-media correlation like coupling newswire with microblogs; however, most of them involve improving tasks on newswire by utilizing complementary information from microblogs, e.g., improving news article summarization using tweets (Gao et al., 2012; Wei and Gao, 2014), generating event summaries through comments (Wang et al., 2015), etc. There is very limited work on using newswire and generating microblogs, e.g., article tweet generation (Lloret and Palomar, 2013) and indicative tweet generation (Sidhaye and Cheung, 2015). Lloret and Palomar (2013) observed that off-the-shelf extractive models produce summaries that have high quantitative scores, but that are not interesting enough. Similarly, Sidhaye and Cheung (2015)'s analysis of indicative tweets shows the narrow overlap between such tweets and their source limits the application of an extractive method for generating them. Our controlled compilation of such tweets shows a mean percentage match of 68.3% (std: 16%) with its source. These analyses strongly suggest that indicative tweets are not regular information-disseminating short texts. Also, the mixed nature of such texts suggests an abstractive, rather than extractive study.

Most abstractive summarization systems use a popular dataset, CNN/DailyMail(Napoles et al., 2012), that includes news articles and story highlights to train and test their performance. However, story highlights are brief and self-contained sentences (about 25-40 words) that allow the recipient to quickly gather information on news stories; it is largely extractive (Woodsend and Lapata, 2010). Our novel corpus includes not only extractive short texts (i.e., story-highlights) and nearly extractive (i.e., headlines), but also very abstractive teasers, and therefore is a challenging and more appropriate dataset to measure abstractive systems.

6 Conclusion

We defined a novel concept of a teaser, a Shortcut-Text amalgamating interesting facts from the news article and teasing elements. We compiled a novel dataset that includes all of the short texts that are associated with news articles. We identified properties like abstractive, teasing, and bona-fide that assist in comparing a teaser with the other forms of short texts. We illustrated techniques to control these properties in teasers and verified their impact through experiments. An overlap-based comparative study of headlines and teasers shows teasers as abstractive while headlines as nearly extractive. Thus, we performed neural abstractive summarization studies on teasers and set a strong benchmark on the novel task of teaser generation.

Acknowledgments

We thank Siemens CT members and the anonymous reviewers for valuable feedback. This research was supported by Bundeswirtschaftsministerium (bmwi.de), grant 01MD15010A (Smart Data Web).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jonas Nygaard Blom and Kenneth Reinecke Hansen. 2015. Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87 – 100.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 93–98. The Association for Computational Linguistics.

Daniel Dor. 2003. On newspaper headlines as relevance optimizers. *Journal of Pragmatics*, 35(5):695 – 721.

Wei Gao, Peng Li, and Kareem Darwish. 2012. Joint topic modeling for event summarization across news and social media streams. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1173–1182. ACM.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1188–1196, Bejing, China. PMLR.

Elena Lloret and Manuel Palomar. 2013. Towards automatic tweet generation: A comparative study from the text summarization perspective in the journalism genre. *Expert Syst. Appl.*, 40(16):6624–6630.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

I. S. P. Nation. 2001. *Learning Vocabulary in Another Language*. Cambridge Applied Linguistics. Cambridge University Press.

M. E. J. Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389. The Association for Computational Linguistics.

Marc Schulder and Eduard Hovy. 2014. Metaphor detection through term relevance. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 18–26, Baltimore, MD. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Priya Sidhaye and Jackie Chi Kit Cheung. 2015. Indicative tweet generation: An extractive summarization problem? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 138–147, Lisbon, Portugal. Association for Computational Linguistics.

Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1055–1065. Association for Computational Linguistics.

Zhongyu Wei and Wei Gao. 2014. Utilizing microblogs for automatic news highlights extraction. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 872–883. ACL.

Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574, Uppsala, Sweden. Association for Computational Linguistics.

A Supplementary

A.1 List of Twitter accounts

The following is the list of Twitter accounts from which data was collected.

Account ID	Account name
759251	CNN
807095	nytimes
35773039	theatlantic
14677919	newyorker
14511951	HuffingtonPost
1367531	FoxNews
28785486	ABC
14173315	NBCNews
2467791	washingtonpost
14293310	TIME

2884771	Newsweek
15754281	USATODAY
16273831	VOANews
3108351	WSJ
14192680	NOLAnews
15012486	CBSNews
12811952	Suntimes
14304462	TB_Times
8940342	HoustonChron
16664681	latimes
14221917	phillydotcom
14179819	njdotcom
15679641	dallasnews
4170491	ajc
6577642	usnews
1652541	reuters
9763482	nydailynews
17469289	nypost
12811952	suntimes
7313362	chicagotribune
8861182	newsday
17820493	ocregister
11877492	starledger
14267944	clevelanddotcom
14495726	phillyinquirer
17348525	startribune
87818409	guardian
15084853	IrishTimes
34655603	thesun
15438913	mailonline
111556423	dailymailuk
380285402	dailymail
5988062	theeconomist
17680050	thescotsman
16973333	independent
17895820	daily_express
4970411	ajenglish

Table 14: To access the Twitter page of an Account name X, use URL https://twitter.com/X and to access the Twitter page of an Account Id X, use URL https://twitter.com/intent/user?user_id=X. The script to download tweets from the above accounts is available in https://github.com/sanjeevkrn/teaser_collect.git.

A.2 Results

The following table shows examples that include input news articles and short text outputs (headline, highlight, and Teaser) both ground-truths and model generations.

Article	Sir Robert Fellowes , the Queen 's private secretary , was on the verge of making the extraordinary request . . .. But he was persuaded to back off by fellow courtiers and the party went ahead as planned putting Camilla . . . . a visible and acknowledged part of the Prince 's life . A new book , The Duchess : The Untold Story by Penny Junor , makes sensational claims about Prince Charles . . . . furious about the birthday party even though by this stage she was fairly relaxed about Charles 's relationship . . .
	Ground-Truth
Headline	Princess Diana latest: Queen aide planned to end Charles affair with Camilla amid rage
Highlight	PRINCE Charles faced being told by the Queen to end his relationship with Camilla after Princess Diana erupted with fury over a lavish party for the Duchess-to-be , it is claimed .
Teaser	**Royal intervention** threatened Charles and Camilla's affair after Diana's **fury** at **posh** bash
	Generated
Headline	duchess of the queen 's private secretary robert fellowes
Highlight	sir robert fellowes , the queen 's private secretary , was on the verge of making the extraordinary request of her majesty .
Teaser	penny junor **reveals** why she was on the **brink** of making the queen 's life
Article	The top Democrat on the Senate Finance Committee asked the Trump administration on Thursday to turn over the names of visitors . . .. That investigation found that members of the golf clubs Trump visited most often as president . . . . Membership lists at Trump's private clubs are secret. USA TODAY found the names of about 4,500 members by reviewing . . . . In a letter to the Department of Homeland Security's Acting Secretary Elaine Duke, Wyden said USA TODAY's examination . . .
	Ground-Truth
Headline	Senator seeks visitor logs, golf partners
Highlight	An investigation by USA TODAY prompts a senior Democratic senator to seek visitor logs at Trump clubs and names of his golfing partners .
Teaser	**Citing** USA TODAY's investigation, a top **Sen.** Democrat **seeks** visitor **logs** to Trump's golf courses & golfing partners.
	Generated
Headline	trump [UNK] to turn over trump 's private clubs
Highlight	the top democrat on the senate finance committee asked the trump administration .
Teaser	the top democrat on trump 's golf club : " it ' s a **lot** of **money** , but it ' s not **going** to

Table 15: The table shows input articles, ground-truth short texts, and seq2seq_pointer generated short texts. Non-overlapping words between short texts and Articles are in bold blue.