Ideas around Content Repurposing

I found content repurposing really interesting and looked at some existing work in the domain.

Talks about our current problem. In order to fit into a tweet or a blogpost, we need to condense/summarize the article.
http://www.abigailsee.com/2017/04/16/taming-rnns-for-better-summarization.html

Abigail See's Paper -
https://arxiv.org/abs/1704.04368

Few Papers on Style Transfer - Summarised content should be in some particular form. We might need to capture dialects of English or of a particular jargon used by a community/age group - finally we fit the style to that of a typical blog post/tweet.
https://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf
https://arxiv.org/abs/1711.06861
https://arxiv.org/abs/1804.06437

The IREL paper for SciBlogger:
http://web2py.iiit.ac.in/research_centres/publications/download/inproceedings.pdf.a292146045c2cd51.63696b6d2d73686f72742e706466.pdf

Twitter Guidelines on How to make effective tweets - can serve as useful metrics for evaluating our model
https://business.twitter.com/en/blog/7-tips-creating-engaging-content-every-day.html

Talks about the common content repurposing tasks out there
https://www.wordstream.com/blog/ws/2015/02/03/repurposing-content

Extensive elaboration on content repurposing
https://neilpatel.com/blog/content-repurposing-whats-old-is-new-again/

A neural network implementation for text-generation
https://github.com/minimaxir/textgenrnn

A project on generating tweets based on the above

https://github.com/minimaxir/tweet-generator

Summarization tools available online
https://www.textcompactor.com/
https://smmry.com/
http://autosummarizer.com/

Flesch Reading Ease (FRE)
https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests

Washington Post's Robot Reporter Heliograf
https://digiday.com/media/washington-posts-robot-reporter-published-500-articles-last-year/

Tweet Generators
http://simitator.com/generator/twitter
http://www.prankmenot.com/
https://www.tweetgen.com/
https://zeoob.com/generate-twitter-tweet/

Some Ideas - Mostly relates to building a product

src -> Encoder hidden states -> partial summary (Decoder hidden states)

Encoder states produce attention distro, a context vector
Vocab distro : Context vector & decoder hidden states
Combine vocab distro and attention distro to produce final disto

Train a covloss - to prevent repetitions.

-Higher level abstraction: kind of unsolved
-Order importance of pieces - how to capture semantically what is more important
-Use of anaphors without antecedents: First reference to an entity made using a pronoun
-Incorrect composition of fragments - Germany beat France morphs into France beats Germany: PROPOSE: use an entailment structure to look for contradictions

Sci-Blogger uses this: Blog title generation - Use of heuristics to produce a partial summary: Attention distro created.
Measures of Performance to keep in mind ->

ROUGE_L: based on LCD b/w candidate & ref
CIDEr: n-gram overlap
Skip Thought Cosine Similarity: Skip thought vectors - continuous representation of sentences - sentence level similarity
**FRE (Flesch Reading Ease): Readability - Avg #words in sentence & #syllables per word are the factors. > 35 : Can be read by high school students.

Extractive tools: Textcompactor, Smmry, autosummarizer: Picks on lines with cues such as "In a nutshell", end of sections, paragraphs, sentences with loads of action words.

Washington Post's Robot reporter Heliograf: Focus on areas where the agency does not want to allot journalists to. Key Ideas:
-> Extend audience producing more #articles
-> Enable journalists to do more high value work
-> Spot trends, pattern recognition
-> Create alerts when some event takes place
-> Pageview matters **

Open source implementations of natural language generation: textgenrnn. Well documented.

Input of a string of chars - convert to 100 D-character embedding vector - sequentially fed to 2 128-cell LSTMs - concat them - All 3 layers fed into Attention layer: Weight most important temporal features, average them together - generate probabilities of upto 394 different characters that they are the next character in the sequence

Other variations exist such as contextual mode - learn text given the context

Applied in Tweet Generator: Train using context labels for tweet synthesis - optimized for tweet generation - any #twitter_users - blend together for hilarity

Many repos doing similar things for Trump tweets

Online tools for fake tweet generation
- Simitator
- Prankmenot
- Tweetgen
- Zeoob

Style Transfer: Sufficient parallel data - Not always available - Work out other methods
- Separate content from style: Tried out on Images before
- Learn separate content & style representations using adversarial networks - Copied src text or replaced a few words
- Two factors are key here
-Transfer strength: LSTM sigmoid classifier - whether it goes to required style or not - Paper <= 0.5, News > 0.5
-Content preservation: Cosine distance between source Vs and target sentence embeddings Vt
-Using cross alignment: Have sentence domains X1 & X2 and Style domains Y1 and Y2 & a latent shared content space.
Encoder E: Map sentence to its content representation
Generator G: Generate sentence back with a style domain
- Poetry generation tried out in 3 papers
- Modern English to Shakespearean English - Had parallel data - seq2seq + pointer generation - Modern shakespeare word dictionary to form candidate words for the pointer generator - it is a scarce resource

Content repurposing - to reach out new audiences, repurpose old content which has the potential of being evergreen, popular, or has received new purpose for some reason.

Ideas:
- Produce guides: Article about gardening. Try factoring it out into key steps
- Lists of things e.g: 12 benefits of an eye-care monitor
(May not explicitly market a brand but show its benefits, let the reader make the connection)
- Process data to produce case studies
- Use of interviews - Credibility increases - Email questionnaires (2 or 3 questions) - Get promoted by the interviewees - Use this for Twitter - Start a trend: What the expert says and include link to main article
- Summarise answers from QnA forums into a blogpost
- Stats & facts: Convert to Twitter posts: Did you know?...
- Pick out tips from large articles: Make a life hack tweet

Content Creation for Tweets - New Directions:

- Mini-campaigns: Separate into Data Points, quotes, analytics from presentations, infographics - include link to long form asset (which may be a blogpost or an article) : can lead to the dev of a themed content series
- GOAL: Drive traffic towards your content
- Polls: Figure out questions & possible options
- Use of visual assets - Let CVIT folks worry about that
- Start new hashtags

Mostly about marketing - getting traffic
- Know which articles/posts received views - have potential for being viral
- Evaluate on parameters:
  - 1500 word sweet spot
  - Headline formula: # JJ Keyword Rationale Promise
    10 proven winter blues remedies
  - Social buttons
- Know what sells when:
  - How to articles
  - Lists
  - What posts
  - Why posts
  - Videos
  - White papers/Case studies

Other Explorable Areas:
- Podcasts
- Instagram
- Infographics
- Ebooks
- Slideshows

Other Things I touched upon:

BERT
https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
Basic ML from the AndrewNG course
Some DL as covered in the tutorial sessions
NLP with DL - glanced through the content to be covered
http://web.stanford.edu/class/cs224n/
https://www.udacity.com/course/deep-learning-pytorch--ud188

Some Datasets I looked at:

EU News Summaries, LDC, Word Clusters, Twitter user income...

1 http://lexhub.org/data_sets/14

EU News Summaries - Downloaded on your desktop
Check the readme - contains all information about the project

This dataset consists of the press summaries of EU news made by the Open Europe think tank (http://www.openeurope.org.uk/Page/PressSummary/en/) in the interval 1st February, 2006 to 15th November, 2013. The press summaries are daily aggregations of news items about the EU or member countries with a focus on current affairs; the news outlets used to compile each summary are listed below the summary's text. The site is updated every weekday, with the major news being covered in a couple of paragraphs, and other less prevalent issues being mentioned in one paragraph to as little as one sentence.

- 'summaries' folder, includes a html page for each news summary

- 'summaries-ann' folder, includes xml files for each summary containing the original html page with GATE (http://gate.ac.uk/) annotations for: POS tags, named entity and DBpedia annotations.

- 'stop-en' file, the list of stopwords used

- 'extract.py' script, use to process the xml files to a vector space representation:
python extract.py summaries-ann 30k/ 30000 10

where the 'summaries-ann' folder is used as input, '30k' is the name of the output folder, 30000 is the number of unigram + bigram features, and 10 the threshold for the number of document/user.

The output folder contains five files:
- dates: the date index
- dictionary: the feature index
- users: the news source index
- sora_vs: 'date_id user_id token_id<TAB>n' where n is the number of occurrences of feature_id in summaries with user_id as the source on day date_id

- sora_vsd: 'date_id user_id<TAB>n' where n is the number of summaries where user_id is the source on day date_id

We can also change the types of features included (by default unigrams and bigrams) by inspecting the python code and following the instructions in the comments.

2 http://lexhub.org/data_sets/11

LDC: Datasets organized by year or projects. Can view the top 10 corpora - gets updated - search feature

-> Concretely Annotated New York Times
It adds multiple kinds and instances of automatically-generated syntactic, semantic and coreference annotations to The New York Times Annotated Corpus (LDC2008T19). all of the 1.8 million articles in The New York Times Annotated Corpus. Those articles were written and published by the New York Times between January 1, 1987 and June 19, 2007

The following layers of annotation were added by processing the articles under the Concrete schema:

- Segmented sentences and Penn Treebank-style tokenized words
- Treebank-style constituent parse trees
- Four different syntactic dependency trees
- Named entities
- Part of speech tags
- Lemmas
- In-document entity coreference chains
- Three different frame semantic parses

Binary Format

For many of these datasets, a particular fee has to be paid for use and access.

3 http://lexhub.org/data_sets/13

Word clusters

The clusters are hard clusters (one word can belong to only one topic).
The importance score represents how central is that word in its cluster.

Clusters are computed using spectral clustering over a word-word similarity matrix.
The difference between methods is how the similarity is computed.
The clusters mainly are useful as features in predictive tasks.

1. NPMI
The word similarity is the Normalised Pointwise Mutual Information over a large corpus. Here, the corpus is 58 days of 10% of the Twitter stream.

2. W2V
Similarity is the cosine between the two word embeddings. The embeddings are derived using Word2Vec with a layer size of 50. The embeddings are learned on the same corpus as the NPMI matrix using Gensim http://radimrehurek.com/gensim/models/word2vec.html

3. GloVe
Similarity is the cosine between the word embeddings obtained using GloVe. The embeddings are downloaded from http://nlp.stanford.edu/projects/glove/
with layer size of 200.

word<SPACE>importance<SPACE>cluster_id

running 0.00083 15
ball 0.00067 15
missed 0.00090 15

4 http://lexhub.org/data_sets/20

Twitter User Income
Feature representation for each of the five feature categories: f-profile, f-shallow, f-demo, f-emo, f-topics

f-demo:
user_id age gender:female_gt_0_5 anxious:agree anxious:strongly_agree anxious:disagree anxious:strongly_disagree anxious:neither children:no children:yes education:degree education:graduate_degree education:high_school excited:agree excited:strongly_agree excited:disagree excited:disagree_strongly excited:neither income:below income:above income:very_high intelligence:above_average intelligence:average intelligence:below_average intelligence:much_above intelligence:much_below narcissist:agree narcissist:strongly_agree narcissist:disagree narcissist:disagree_strongly narcissist:neither optimism:extreme_optimist optimism:extreme_pessimist

optimism:neither optimism:optimist optimism:pessimist political:conservative political:independent political:liberal political:unafiliated race:asian race:afr_american race:hispanic race:indian race:other race:caucasian relationship:divorced relationship:in_a_relationship relationship:married relationship:other relationship:single religion:christian religion:hindu religion:jewish religion:muslim religion:other religion:unaffiliated

Probabilities for each.

f-emo:
user_id positive neutral negative joy sadness disgust anger surprise fear

f-profile:
user_id followers friends listed foll_friend_ratio favourites tweets_day tweets enlish_tweets

f-shallow:
user_id non_duplicate retweeted retweets_tweet retweets hashtags tweets_hashtags tweets_at_mentions at_replies unique_at_mentions urls

f-topic:
User_id Topic_1 Topic_2 … Topic_200

Users Income
user_id mean_income

Word-topics-200
Topic Word Centrality

5191 users annotated with their income through their SOC 3-digit occupation.

Archive contains:
Feature representation for each of the five feature categories: f-profile, f-shallow, f-demo, f-emo, f-topics
Users mapped to their income: users-income, Words for each topic: word-topics-200