International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 4, ISSN No. 2455-2143, Pages 197-202
Published Online August 2019 in IJEAST (http://www.ijeast.com)

CONCISENET: AN END TO END ABSTRACTIVE MODEL FOR TOPIC GENERATION

Saurav Saha
Department of CSE
SRM IST - KTR
Chennai, India

Abhilash Pal
Department of CSE
SRM IST – KTR
Chennai, India

R. Anita
Department of CSE
SRM IST - KTR
Chennai, India

Abstract - Recent approaches in Title generation using neural approaches have relied on an end to end deep learning system based on the sequence to sequence model. Such approaches have yielded good results but remain constricted in use due to a fixed size input which is often very small compared to the text being used or might take huge compute power to train and use if input size is increased. Our approach amalgamates an extractive and abstractive approach to get the best of both worlds using a textrank algorithm for the extractive part and a reasonably small seq2seq architecture as the abstractive part. Testing on the Amazon Fine Food Review dataset, our approach gives good results using less compute power.We utilize the prevailing metrics of ROUGE and Cosine Similarity. Manual checking shows that the majority of our generated topics are grammatically correct.
Index Terms—Deep learning, Machine learning, Predictive models, Natural language processing, Tokenization

I. INTRODUCTION

Text Summarization in recent times has employed either an extractive or abstractive methodology. The former may utilize a neural approach as in [4] or utilize a graph based ranking algorithm as in [7] and related works. Abstractive methods are also gaining prominence in recent years with breakthroughs in Deep Learning and availability of datasets and computation power and resources. Such abstractive methods can be either based on a sequence to sequence model or a basic LSTM model which suitable tackles the problem of many to many mapping. Our contribution links both these methods, utilizing the speed and ease of the Textrank algorithm[7] alongside the robustness of deep learning models. As the internet grows in size and use, we see a explosion of data. A huge part of this data is text, which grows in the form of news articles and blogs. Indexing this huge amount of textual data is extremely important. Our approach can handle large amounts of data, due to our usage of the Textrank algorithm[7] and generate a proper short summary or topic, utilizing the very best in current deep learning research. The paper is divided into five sections. Section II explains the current trends in research, and how we build our own work on top of that. Section III explains our model, it's various parts and the training parameters, The next section IV defines the evaluation metrics and showcases a few outputs for us to evaluate our workagainst. Section V concludes the paper with added anecdotes about how successive research may be handled.

II. RELATED WORK

Summarizing text requires an innate amount of insight over a particular piece of text and has been used widely in Natural Language based applications. Text Summarization can be of two major types extractive and abstractive. Extractive Text Summarization provides a summary on a piece of text on the basis of sentences already present in the input text and merely uses an algorithm to chose sentences on the basis of their importance to be used in the summary. In case of Abstractive Text Summarizers, capture the information of the whole piece of text input and create an entirely new piece of text containing new sentences. Sentences are not extracted but generated in this case.

A. Past Work on Extractive Text Summarization

Textrank is one of the most popularly used algorithms that have been used up till date for text summarization, not using a deep learning approach. It was introduced in 2004 by Mihalcea and Tarau in Textrank: Bringing Order into texts [7]. It used an unsupervised approach and is still utilized in text summarizers which are extractive in nature to tackle the situation by providing a network graph based solution. It works by asserting the importance of a specific node over others in the system. In case of a text summarization task, Textrank[7] arranges the sentences in a descending order of their similarities to each other to capture the most amount of information from the piece of text to generate a summary. The Textrank algorithm was inspired by Google's PageRank algorithm which was able to bring about a revolution in the field of web search technology.

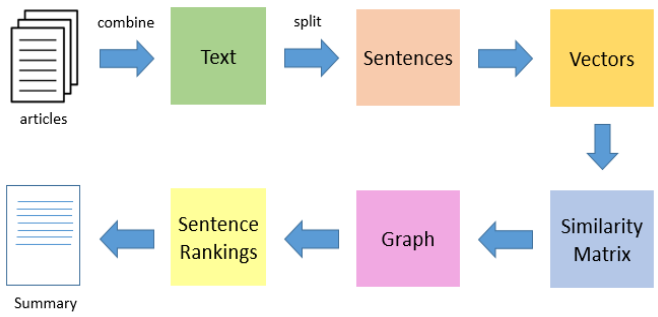

Fig.1. TextRank Overview [11]

Extractive Text Summarization can also use Deep Learning based approaches where sentences are treated as long sequences of text requiring classification. SummaRuNNer by Nallapati, Zhai and Zhou[4] used a

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 4, ISSN No. 2455-2143, Pages 197-202
Published Online August 2019 in IJEAST (http://www.ijeast.com)

Recurrent Neural Network(RNN) based sequence model for summarization of documents. The model could be abstractively trained on an extractive model that could be trained on human generated reference summaries alone, which eliminated the requirement for sentence level extractive labels. SummaRuNNer used a GRU based Recurrent Neural Network which was trained on the CNN/ DailyMail corpus created originally for question answering which was based on passages and was modified for the task of document summarization. The model also used the DUC 2002 single document summarization dataset which consists of 567 documents to evaluate model in an additional external domain test.

BERT or Bidirectional Encoder Representations from Transformers is a State of the art language model for Natural Language Processing which was introduced by researchers at Google AI Language. BERTSUM[13] uses bert-base-uncased version of BERT which is a pre-trained Transformer model, which has achieved substantial performance of NLP based applications. BERTSUM[13] is a simple variant of BERT used specifically for extractive text summarization tasks. The model for BERTSUM was trained majorly on the CNN/ Daily Mail dataset[14] and the New York Times[15] Annotated Corpus which outperformed the last best performing system by 1.65 on the ROUGE-L metric.

Reinforcement Learning can also be used in Text Summarization as done in [17] where it had been used to globally optimize the ROUGE metric for evaluating. The model specifically used Reinforcement Learning to rank sentences for generating a summary for an extractive text summarization implementation.

B. Past Work on Abstractive Methods

Abstractive Text Summarization requires the use Deep Learning based approaches where the overall knowledge and information of the piece of text needs to be captured. For example, Abstractive text summarization using sequence-tosequence RNNs and beyond by Nallapati, Zhou, Santor, Gulcehre, Xiang[5] which used an Attentional Encoder-Decoder Recurrent Neural Networks based model as its architecture which addressed some key problems generally faced in text summarizing by modelling keywords, containing sentence to word structure hierarchy.The model used a bidirectional GRU-RNN based encoder and a uni-directional GRU-RNN based decoder architecture, which is unlike our model which uses an LSTM-RNN based encoder and decoder. The model was trained on the annotated GigaWord Corpus, DUC Corpus and the CNN/ Daily Mail Corpus and was evaluated on metrics such as Rouge-1, Rouge-2 and Rouge-L. In Deep Reinforced Model for Abstractive Summarization by Paulus, Romain, Xiong and Socher [2], a Reinforcement Learning model with intra-attention is used which although computationally expensive, gives great results.

Pointer Generator Networks can also be used to create abstractive text summarization models such as the one used in Get To The Point by See, Liu and Manning[16]. A hybrid pointer-generator network was used that had the capability of copying words from a reference text through pointing, which aided to the accuracy in reproduction of knowledge, while also retaining the ability to generate new words via the generator. Coverage was implemented to track the content that had been summarized which penalized repetition. The model was trained using the CNN/ Daily Mail[14] Dataset and outperformed the previous state of the model by at least 2 ROUGE points.

Extreme Summarization by Narayan et. al.[18] uses a Convolutional Sequence-to-Sequence Learning Model for abstractive text summarization exclusively conditioned on the topic of newspaper articles. The model was trained over the XSum dataset which was created using BBC articles and their single line sentence summaries.

III. PROPOSED WORK

We introduce three models for abstractive summarization CNet, CNet+G, CNet+2G. CNet features the basic architecture where the embedding layer is trainable in both the encoder and decoder. This lets the model learn it's own vectorized representations of words that it encounters during training. The next model CNet+G utilizes pretrained GloVe vectors[10] in the encoder part. GloVe is a pretrained vector representation of common words in the english vocabulary. We use 100 dimension GloVe vectors, but available vectors range from 50 dimensions to 300 dimensions. This means that a big part of parameters is not trainable and this speeds up training speeds. The model CNet+2G utilizes pretrained GloVe vectors in both the encoder and decoder part. This further speeds up training. Comparison of the three models is given in Table I.

A. Basic Architecture

The main model architecture consists of 2 distinct components, the first component is the extractive model which is fed with a large paragraph or text and supplies the output in the form of the top n sentences according to the provided input. The extractive model compares each sentence with every other sentence by considering the vocabulary, which is the number of unique words present in both the sentences, in the text. Stopwords in the sentences are not considered in the vocabulary as they are inconsequential in our case.

Vectors of similar dimensions are created on the basis of the vocabulary and hence the amount of similarity between 2 sentences is measured and a score is provided on the basis of cosine similarity between both the sentences. Cosine Similarity measures the similarity of 2 specific non zero vectors of an inner product entity that measure the angle of cosine between them, similar to how a dot product is generated for 2 vectors. A Similarity Matrix is then generated with respect to the individual similarity scores provided for each of the sentences from the text that are compared. Using this Similarity Matrix, a similarity graph is generated by using the Text Ranking algorithm[7].

Text Ranking does not require any training data and can be used on any random piece of text and is a ranking algorithm specifically based on a graph network. The nodes on the graph signify the each of the sentences in the text provided as input and the distance between each of the nodes is specified with respect to the similarity score generated for both the sentences. If two particular sentences are very similar to each other, then this signifies that the distances between the nodes signifying them will have a smaller distance and a higher similarity score when compared to two other nodes on the graph which have a smaller similarity score specifying, that they have a smaller similarity score and hence the distances between the node will be certainly larger. On the basis of this network graph generated with respect to the similarity scores of sentences, sentences are ranked on the basis of their similarities.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 4, ISSN No. 2455-2143, Pages 197-202
Published Online August 2019 in IJEAST (http://www.ijeast.com)

In this case, ranking of the sentences is done on the basis of how dissimilar 2 specific sentences are, this is because the text is being summarized on the basis of the sentences that are being extracted. The aim is to gather the gist of the entire piece of text and hence requires the use of sentences which are unique to one another. Hence, the sentences are ranked in a decreasing order of their similarity. From this new list of sentences the top N sentences capable of capturing the knowledge of the whole text are selected and a summary is then generated from these N sentences which we assume to be not larger than 80 words.

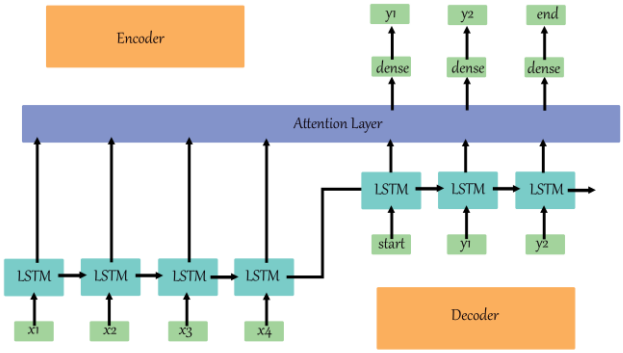

Fig.2. Main Model Architecture.

B. Modules

- Encoder

  The Encoder uses a multiple layers of a network to convert the input sentences into the form of their corresponding hidden vectors. These vectors are either being trained as the model trains as in CNet, or utilize pretrained GloVe Vectors [10] as in CNet+G and CNet+2G. The encoder LSTM parses the entire sequence of input, one element at a time at each time step. It is then able to process all the relevant information to capture the contextual knowledge present in the entire sequence of the input provided at each and every timestep. In this case the Encoder takes in an input of less than or equivalent length of 80 elements and its output is a hidden vector having 500 elements or dimensions. The encoder uses a triple layer stacked lstm architecture whose output is fed into the attention layer of the architecture.

- The Decoder is a component which uses an LSTM architecture. It receives the entire hidden vector and using it, generates the ouput word by word at each an every instance or a timestep. The hidden vector that the decoder uses as input is first injected with a START token at the beginning of the sequence and the an END token is added at the end of the sequence. While decoding this vector, the target sequence is unestablished. The Decoder starts predicting the target sequence by first parsing the START token which signifies the beginning and the end is denoted by the ENDtoken, after which it understands that there are no more words to be processed. Each recurrent unit of the LSTM architecture accepts an element from the hidden vector also known as a hidden state from the previous unit and also produces anewhiddenstateas well as its own hidden state. The architecture is presented in Fig. 2.

C. LSTM Unit Overview

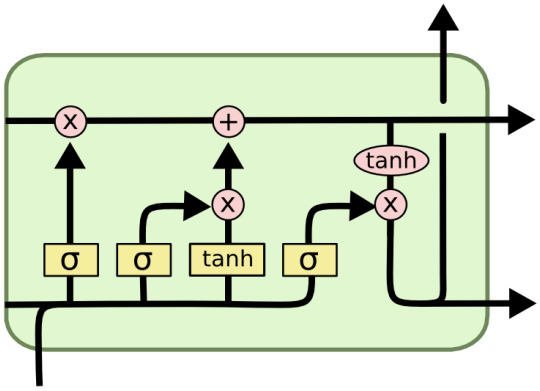

Fig.3. LSTM Unit [8].

Our model uses three layers of stacked LSTM(Long Short Term Memory) units as the Recurrent Neural Unit. LSTMs have gained a lot of traction in recent years due to the way they handle the vanishing gradient problem with ease. Our models were trained for about ten epochs each with RMSProp as the optimizer. Using RMSProp instead of traditional optimizers like Stochastic Gradient Descent improves training efficiency and helps the model converge better.

The LSTM layer consists of three distinct gates - the Input gate, Forget gate and Output gate.The Input gate informs what newinformationweregoingtostoreinthecellstate,theForget gate tells us about the information to throw away from the cell state. The output gate is used to provide the activation to the final output of the LSTM block at a given timestamp.

D. Attention Layer

The attention layer sits on top of the encoder and decoder LSTM layers. Through training it learns to concentrate more on parts of the output of the previous layer which is mostly required for generating the summary in the next layer using the dense output layer. Without the attention layer, the encoder would have to pass on the entire information about the input sequence using the thought vector, leading to information loss and ambiguity. The attention layer, as shown in Fig. 2 solves this issue by taking information from the encoder at each step and weighing them in accordance to the decoder's needs.

E. Dataset

We train and test our model on the Amazon Fine Food Reviews dataset [9]. It consists of 100000 reviews of products on amazon alongside their short summaries in phrases. Apart from this, it also contains a helpfulness indicator and a rating index in the range 1 - 5. After removing rows for missing data and duplicates, we end with 88421 rows. This is further divided into testing and training sets. The training set consists of 79516 data points while the test set consists of 8836 data points. Our evaluation given in Table II includes 5 random sentences from the test set.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 4, ISSN No. 2455-2143, Pages 197-202
Published Online August 2019 in IJEAST (http://www.ijeast.com)

F. Comparison of Presented Models

CNet utilizes around 56M parameters, with each epoch taking around 22 minutes to train. CNet+G has 34M parameters out of which 28M are trainable(the rest accounting for the encoder embedding layer) and it takes around 20 minutes for each epoch. CNet+2G has 28M total and 21M trainable parameters but epochs take around 30 minutes to train. Our models were trained on Google Colab's environment using the NVIDIA Tesla K80 GPU accelerator. It offers 24GB memory for training the model.

IV. EVALUATION

A. Metrics

We utilize Cosine Similarity and ROUGE as methods to evaluate how well our models are doing.All models get similar scores on Cosine Similarity metrics while the ROUGE metrics show a bit of variation in the scores. Table I showcases how well our models perform on the Amazon Fine Food Reviews dataset[9].

The Cosine Similarity works by forming a vectorized Bag of Words representation of the ground truth label and the outputs we get from the models.

$$CosineSimilarity(x, y) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}$$

Here x and y refer to the two vectors formed out of the two sentences being compared. The vectors are formed using Bag of Words, which counts the frequency of unique words in both the sentences. In the numerator we have the dot product of x,y and in the denominator, the absolute values of x and y respectively are multiplied.

Recall Oriented Understudy for Gisting Evaluation or ROUGE is an evaluation metric for specifically evaluating Text Summarization and Machine Translation application on the basis of metrics such as Precision and Recall.

Recall describes the amount of information being captured from a piece of reference text by providing a ratio between the total amount of overlapping words in the reference text and the summary versus the total words in the piece of reference text.

$$Recall(x) = \frac{P(x)}{Q(x)} \quad \text{and} \quad Precision(x) = \frac{P(x)}{Q'(x)}$$

P(x) = no. of overlapping words
Q(x) = total words in reference summary
Q'(x) = total words in system summary

Recall tells us about the amount of information that has been captured in the generated summary but this information might not be necessarily relevant. This is a particular downfall of the Recall metric and hence Precision as a metric is used to measure the amount of relevant information that is captured from the piece of reference text. Simply put, Precision is the ratio of number of overlapping words present in both the summary generated and the reference text versus the total words that are present in the generated summary.

Rouge-1 is an evaluation metric which describes the overlap of unigrams in the output summary and the reference summary. Rouge-2 on the other hand is an evaluation metric focussing on the overlap of bigrams in the output summary and the reference summary. Rouge-1 is generally used over or together with the Rouge-2 metric to show the fluency in the summary.

Rouge-L is to measure the amount of longest matching sequence of words using longest common subsequence or LCS. The utility of using LCS is that it doesnt need matches consecutively but in-sequence matches that specifically describe sentence level word order. It natively includes in-sequence longest matching N-grams and hence does not require the explicit definition of an N-gram length.

B. Results

TABLE I
EVALUATION SCORES

Model	Cosine Similarity	ROUGE-1	ROUGE-2	ROUGE-L
CNet	0.135	13.32	2.8	11.89
CNet+G	0.135	12.85	2.61	11.43
CNet+2G	0.133	13.02	2.60	11.58

According to Cosine Similarity, all three models offer the same performance. This can be misleading at times as if different words with similar meaning are compared, cosine similarity will be low while information conveyed may be the same. The ROUGE scores offer a better way to evaluate how well our models do.

Our models accurately predict the required topic in accordance with the context of the text in question. We have presented the output of five random sentences from the validation set and the output given by each of our three models. This can be viewed in Table II. Manual overview reveals that though the topics generated by our model is a bit generalized at times, it captures the overall gist very effectively.

V. CONCLUSION AND FUTURE WORK

Future work can focus on training bigger networks, capable of outputting longer sentences. A weight factor maybe introduced to penalize shorter sentences, thus forcing the model towards longer and more coherent summaries. Models with alternative recurrent units like GRUs may also be used as in SummaRuNNer by Nallapati, Zhai and Zhou[4].

Research can also focus on better extractive methods, using a neural approach, which trades runtime for better extractive results. An alternative dataset may also be used.

Instead of GloVe Embeddings, BERT Embeddings[12] can be used. These Bidirectionally trained embeddings have shown a lot of potential in recent months. Also, instead of utilizing 100 dimension GloVe Vectors, we can use smaller or bigger, ie. upto 300 dimension GloVe Vectors.

VI. CONCLUSION

Future work can focus on training bigger networks, capable of outputting longer sentences. A weight factor maybe introduced to penalize shorter sentences, thus forcing the model towards longer and more coherent summaries. Models

with alternative recurrent units like GRUs may also be used as in SummaRuNNer by Nallapati, Zhai and Zhou[4]. Research can also focus on better extractive methods, using a neural approach, which trades runtime for better extractive results. An alternative dataset may also be used. Instead of GloVe Embeddings, BERT Embeddings[12] can be used. These Bidirectionally trained embeddings have shown a lot of potential in recent months. Also, instead of utilizing 100 dimension GloVe Vectors, we can use smaller or bigger, ie. upto 300 dimension GloVe Vectors.

VI. ACKNOWLEDGEMENT

The authors thank Dr. Subalalitha C.N, Dept of CSE, SRM Institute of Science and Technology for her support and guidance during this project. We would also like to acknowledge the help we have received from Christopher Olah's Blog on LSTMs[8] during this project.

VII. REFERENCE

[1] Moratanch N. and Chitrakala S. 2017. "A survey on extractive text summarization" in International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai,(pp. 1-6). doi: 10.1109/ICCCSP.2017.7944061

[2] Paulus, Romain, Xiong Caiming and Socher Richard. 2017. A Deep Reinforced Model for Abstractive Summarization. in ArXiv abs/1705.04304

[3] See, Abigail, Liu Peter J. and Manning Christopher D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. in ACL 2017.

[4] Nallapati Ramesh, Zhai Feifei, and Zhou Bowen. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Association for the Advancement of Artificial Intelligence(2016).

[5] Nallapati Ramesh, Zhou Bowen, dos Santos Cicero, Gulcehre Caglar, and Xiang Bing. 2016. Abstractive text summarization using sequenceto-sequence RNNs and beyond. In Computational Natural Language Learning.

[6] Chopra Sumit , Auli Michael , and Rush Alexander M . 2016. Abstractive sentence summarization with attentive recurrent neural networks. In North American Chapter of the Association for Computational Linguistics.

[7] Mihalcea, R., Tarau, P. 2004. Textrank: Bringing order into texts. In Lin, D., Wu D. (Eds.), Proceedings of EMNLP 2004, pp. 404411 Barcelona, Spain. Association for Computational Linguistics

[8] Christopher Olah, 'Understanding LSTM Networks', 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed: 20- August- 2019].

[9] Stanford Network Analysis Project, 'Amazon Fine Food Reviews', 2016. [Online]. Available: https://www.kaggle.com/snap/amazon-finefood-reviews [Accessed: 20- August- 2019].

[10] Pennington Jeffrey, Socher Richard, Manning Christopher D. 2014. 'GloVe: Global Vectors for Word Representation', [Online]. Available: https://nlp.stanford.edu/projects/glove/ [Accessed: 20- August- 2019].

[11] Joshi Prateek. 2018. 'An Introduction to Text Summarization using the TextRank Algorithm' [Online]. Available: https://www.analyticsvidhya.com/blog/2018/11/introduction-textsummarization-textrank-python/ [Accessed: 20- August- 2019].

[12] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.

[13] Yang Liu. Fine-tune BERT for Extractive Summarization. 2019 in arXiv:1903.10318 [cs.CL]

[14] See Abigail. CNN-dailymail dataset. [Online]. Available: https://github.com/abisee/cnn-dailymail [Accessed: 20-August2019].

[15] Sandhaus Evan. The New York Times Annotated Corpus. 2008 [Online]. Available: https://catalog.ldc.upenn.edu/LDC2008T19 [Accessed: 20-August- 2019].

[16] See Abigail,Liu Peter J., Manning Christopher D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. 2017 in arXiv:1704.04368 [cs.CL]

[17] Narayan Shashi , Cohen Shay B., Lapata Mirella. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning in Proceedings of NAACL-HLT 2018, pages 17471759.

[18] Narayan Shashi, B. Cohen Shay , Lapata Mirella. 2018. Dont Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 17971807

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 4, ISSN No. 2455-2143, Pages 197-202
Published Online August 2019 in IJEAST (http://www.ijeast.com)

TABLE II
SAMPLE OUTPUT

Original Text	Original Summary	CNet	CNet + G	CNet + 2G
I ordered the Salmon on Thursday, January 11th and we received it on January 19th. The Salmon was delicious. The wooden box has a nice design and can be used to store items in the future.	alaska smokehouse smoked salmon	great	great value	excellent
I drank this cold and could not be more pleased. The coffee was high quality arabica - I always notice when coffee is arabica or robusta. It was sweet without being too sweet - how'd they do that? I mean, you can't compare this to the major bottled coffee brand for that reason alone. You can keep this in the fridge at work and offer it to people without looking like you're pushing calories along with caffeine.For a long time, I'd stock highly sweetened coffee beverages in the fridge at work. Then I noticed that people stopped consuming them, and it happened around the same time that I lost interest in that level of sweetness. I recommend this drink for those that prefer a lower level of sugar and for old school arabica coffee types.	it was perfect little sweet without being too sweet	great coffee	good coffee	good coffee but not for me
I had a variety of granola. And this one ranks of my top most favorites. It is a very crunch granola with a coconut main taste. It is sweet but not overly so. I eat at as a snack when im craving something. Fills you up very fast and satisfies any sugar cravings.	great taste	great tasting granola	great tasting	yummy
I bought these at the Dollar Tree! (3 oz bags) They are yummy, taste like french fries but crunchy. I going to buy in bulk if they pass the kid test!	yum	great taste	yummy	yummy
If you are used to eating flaxseed, then this is the brownie for you. Hodgson Mill brownies are super easy to make, and taste great. Since I like dark chocolate, I usually add a little more cocoa.	delicious brownie	great for pancakes	great brownies	great for the best