OCCAMS EVALUATION

In the creation of a TLDR for Intelligence Analysts, there is a purported value in developing multi-document summarization techniques to support the synthesis and identification of new, relevant information. One such technology, OCCAMS (an Optimal Combinatorial Covering Algorithm for Multi-document Summarization), aims to do just this using statistical techniques with simplistic hyperparameters to define what 'relevant' information is.

The following report looks to identify the techniques used in evaluating OCCAMS, describe the opportunities for further summarization improvements, and conclude the usefulness in continuing to develop summarization models for the purpose of an intelligence TLDR.

OCCAMS OVERVIEW

OCCAMS has continued to be developed at SCADS 2022. The repo can be found at https://github.ncsu.edu/SCADS/Occams.

Key System Components

OCCAMS takes a selection of text bodies that form a corpus and summarizes them based on a user designated scheme. Within the corpus, 1 to n *documents* can be present, each comprising of *sentences* to be evaluated for relevance. After tokenizing the corpus, an *Incidence Structure* is generated to hold the listing of sentences in the corpus and their associated term weights. Summaries are extractive; thus, the algorithm ranks sentences based on the scheme provided and then selects the highest weighted sentences for use in a summary. Details regarding the full algorithm are described here: https://ieeexplore.ieee.org/document/6406475.

Summaries can then be generated from the Incidence Structure into *Extracts*. These extracts are built using a particular scheme which acts as a heuristic to judge relevance of a sentence. The typical pipeline for sentence generation follows as such:

```
# generates a processors, evaluates texts in corpus to build Incidence Structure
docprocess = DocumentProcessor(TermOrder.BIGRAMS, language='english', download=True).process
documents = [docprocess(str(i)) for i in df['text']]
doc_incidences = IncidenceStructure(documents)

# gathers term weights using a particular scheme/heuristic
extractor = TermFrequencySummaryExtractor.from_documents(documents,
units=SummaryUnits.WORDS, scheme=chosen_scheme)

# extracts sentences of relevance for summary based on a particular length
extractor.extract(budget=length)
```

For more examples on running summarization using OCCAMS visit https://github.ncsu.edu/SCADS/Occams/tree/master/notebooks. Additional notebooks with examples can be found in the LAS Data Science Image within the efs drive. (~/efs/home/pcorona_content)

Schemes – Heuristics of Relevance

As mentioned, relevance is arbitrary. OCCAMS has developed 13 schemes defining how to evaluate which sentences to extract for a summary. The schemes include:

- Term Frequency Scheme: computed based on the normalized count of the terms throughout the full corpus of documents to be summarized.
- Positional First Scheme: increases term weights to prioritize terms present within the first sentence of any document in the corpus.
- Positional Dense Scheme: term weights higher for terms earlier in each document, logarithmic decline in weight as sentences get further toward the end of the document.
- Core Sentence Scheme: prioritizes terms that exist within sentences that have high coverage of the entire document.
- Core Term Scheme: prioritizes terms that that exist within sentences that have high coverage of the entire document while still considering overall term counts.
- Logarithmic Counts Scheme: terms are logarithmically weighted based on their overall frequency within the corpus.
- Counts Scheme: terms are equally weighted based on their overall frequency within the corpus.
- Entropy Scheme: probability distribution across terms within the corpus.
- Flat Scheme: all terms are given equal weight regardless of frequency or position.
- Fisher Scheme: utilizes Fisher Term Weights, computed based on comparing corpus against a background corpus for similarities.
- Positional Minimum/Maximum/Mean Scheme: NA (note: no intuitive rationale for these schemes have been determined.)

These schemes are designed and passed to the summary extractor in the term_weight_schemes.py file within the OCCAMS/summarize folder.

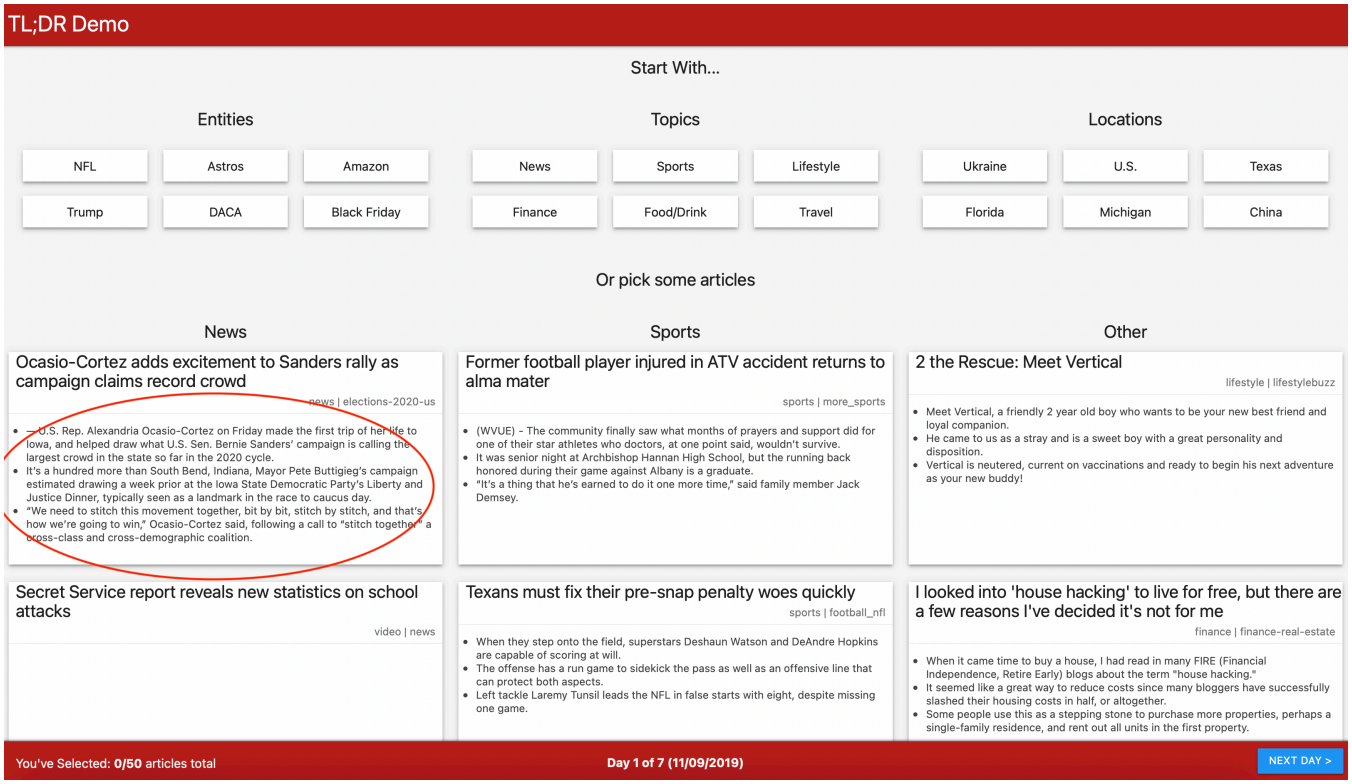
Throughout the exploration of the package, there were some key discoveries that are of value to note:

> While many of the schemes to calculate sentence relevance are independent from document or sentence ordering, it should be noted that the Incidence Structure has essentially collapsed the full corpus into one single text body to be summarized. The result is that summarization occurs as if the corpus is only one document, ignoring the relative clustering of information within documents. When using positional schemes, term weight may be adjusted based on their relative placement within the document, but no relevance is given to the fact that term A and B occurred at the beginning of one document while term C and D were placed in another.
>
> There is no user input to guide summarization mechanisms toward subjects of relevance. Fisher term weights have been utilized to attempt to guide the mechanism in this way, but in general this produces summarizations more similar to known information rather than expanding the scope to include new information included within a corpus.

OCCAMS SUMMARIZATION RESULTS

Single Document Summarization

For the SCADS recommender system demo, single document summarizations (aka highlights) were generated to show an additional aspect of development that occurred during SCADS.



Each of these highlights are ranked by their 'relevance' in summarizing the information within the document. However, they are noticeably non-sequitur when displayed in a paragraph format. This tends to be the case throughout most generated extractive summaries in produced. OCCAMS doesn't reduce this problem any further.

Multi-Document Summarization

Despite this implementation, prioritization was placed on identifying if OCCAMS could produce summaries which effectively shared information across different documents. Ideally, corpuses could be clustered, and one summary could be produced that had high coverage of the entire corpus. Use cases could include summarizing clusters of documents that were recommended or inverting the process to utilize high-scoring summaries as the recommender system's input to selecting new documents.

Summarization Examples

3 clustered documents regarding the University of Connecticut's Women's Basketball team using the Position scheme:

Extracted Sentence	Title of Associated Document
1) The UConn women's basketball team wraps up its slate of exhibition games with a matchup against Trevecca Nazarene Wednesday night at XL center. 2) Crystal is going to have to do that every game. 3) ——— ©2019 The Hartford Courant (Hartford, Conn.) Visit The Hartford Courant (Hartford, Conn.) At www.Courant.Com Distributed By Tribune Content Agency, LLC 4) UConn women's basketball fans got their first look at the 2019-20 Huskies, including three of the program's newcomers, in a 103-40 exhibition win over Division II Jefferson Sunday afternoon at Gampel Pavilion 5) Junior Megan Walker, named all-conference first team, is capable of a breakout season.	1) Mike Anthony: The UConn Women Will Be Good If Crystal Dangerfield Can Be Great 2) Mike Anthony: The UConn Women Will Be Good If Crystal Dangerfield Can Be Great 3) Freshman Anna Makurat starts, Megan Walker shines as UConn women throttle Division II Jefferson 103-40 in exhibition 4) Live updates: UConn women take on Trevecca Nazarene in XL Center exhibition 5) Mike Anthony: The UConn Women Will Be Good If Crystal Dangerfield Can Be Great

145 clustered documents regarding the basketball player Zion Williamson:

Extracted Sentence	Title of Associated Document
1) Who will be in the Final Four? 2) Related Slideshow: Best of the NBA season (provided by imagn) 3) The Houston Rockets meet the New Orleans Pelicans for the second time this season. 4) And that's going to have to be enough. 5) He's got to play the game. 6) He's still on the team. 7) This is a team that nearly made the playoffs last season. 8) This would be the season to do it. 9) Michigan State is the preseason No. 10) Shai Gilgeous-Alexander had 23 points and eight rebounds, and the Thunder beat the Pelicans 115-104. 11) It might take a lot of them.	1) The top 10 questions for the 2019-2020 college basketball season 2) Winners and losers from NBA's opening 3) Houston faces New Orleans after Harden's 42-point showing 4) Expect the Suns to be a lot better this season, but so what? The West is impossible 5) 3-pointers: Takeaways from Rockets' win over Pelicans 6) Lonzo Ball says Pelicans have to 'hold the fort down' until Zion Williamson returns 7) 6 overreactions to the first week of the NBA season 8) Luke DeCock: With talent drain at the top, it's a wide-open ACC basketball race (for a change)) 9) The top 10 questions for the 2019-2020 college basketball season 10) Pelicans drop to 1-5 after loss at OKC 11) There are no winners in NBA-China dustup

As the number of grouped documents increases, summarization generally declines (although this in part changes when utilizing different OCCAMS schemes as well) The goal was to maximize the size of the document's clusters and the summarization quality. We looked to produce intuitive summarizations with high, non-duplicative information coverage. While single document summarization produced less-than-human quality responses, the hope was that the density of the term weight matrix would increase when OCCAMS was provided with more input data. Thus, making multi-document summarization higher quality than single document summarization.

To effectively identify an optimized corpus size, evaluation criteria were identified. These criteria could be utilized in the future to compare OCCAMS with additional novel summarization techniques. However, while measurements of information gain, coverage and novelty can be taken, no evaluation criteria have been identified to quantify the readability of a particular summary. Human evaluation would still be needed in this regard to identify if a summary of 'highlights' would appropriate qualify as a paragraph-based summary. This is hypothesized to be an unachievable goal utilizing extractive text summarization due to its non-generative nature. Abstractive methods can be appealing to overcome this challenge but have been disregarded for TLDR use cases due to hallucinogenic tendencies.

EVALUATION CRITERIA

Primitive evaluation metrics were developed to be used in determining OCCAMS usefulness and capability of being deployed to summarize large corpuses in real time.

Evaluation Metrics – Utility

One measurement of utility comes from document coverage. Summaries would be expected to return knowledge from across documents while reducing duplicative information shared. To measure coverage, entities (person, organization, location) were extracted from both the document(s) and the associated summary using Spacy's *en_core_web_sm* pipeline. A similarity score and average contribution metric were then evaluated. The *similarity score* identified the entities described within the summary as a fraction of entities available throughout the document(s). Summaries with high similarity scores would intuitively contain a higher amount of information from throughout the document(s). The *average contribution metric* measured the distribution of entities across the documents. A higher average contribution would point to improved synthesis of information. It is important to note that this contribution metric can be rendered useless if the size of the summary that the user allows is not large enough to grab sentences from each document in the corpus.

Traditional metrics for OCCAMS have previously been evaluated (see https://ieeexplore.ieee.org/document/6406475). Our goal is not to identify a model comparison technique in this experiment, but rather identify attributes of the summary which we can use to determine if the information gain present within a summary is an accurate and usable reflection of the corpus.

Evaluation Metrics – Feasibility & Scalability

The scalability of our summarization model must also be considered. We evaluated OCCAMS to take an average of .16 seconds length to process and summarize the text of a corpus of length 10. While summarization (once sentences are scored) becomes scalable – one such trial of over 30,000 documents still summarized in under 5 seconds – the process of scoring sentences is time intensive. This same corpus took over 2 hours to build out sentence scores based on the computed term weights. For implementation in interactive applications, there is a need to ensure that any corpus generated (either manually as we did, or automatically through other modelling techniques) is scaled to a size that allows for efficient summarization.

Excluding the Fisher scheme – which requires the computation of document bigrams to be compared against the bigrams of a background corpus, a time-consuming algorithm which demands precomputing and caching for efficient results – different scheme selections do not drastically change the summarization time. For each summarization task, *term weight extraction time* and *summary extraction time* were collected. This effectively breaks up the time that OCCAMS spends in the NLP realm, gathering information regarding the term weights, from the computation time needed to evaluate the relevance of each sentence and rank them for summarization. Thus, using OCCAMS in real time remains reliant on the ability to efficiently process incoming documents; summarization time is negligible.

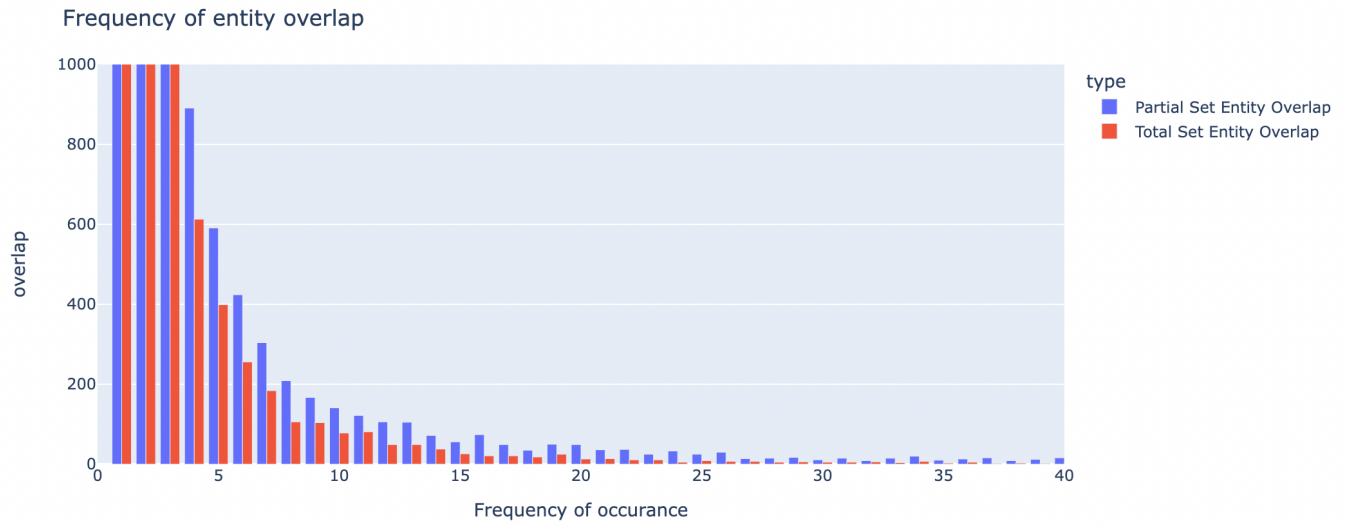
EXPERIMENT METHODS

** To rerun the experiment, follow the INSTALL.md guide to setup the experiment and run common sequences of summaries. **

Defining the Corpus

To identify natural corpuses to split documents into, data was attempted to be clustered based on natural patterns within the text documents – primarily entity overlap.

The MIND dataset had pre-labelled data regarding entities present within the text body. These entities included geographical elements, organizations and named persons and were extracted from the title of the document to provide an umbrella of the documents content. Utilizing these entities, the overlap was found to be minimal (in both partial and total set overlap). With over 91% of the data having less than 6 key terms that overlap with additional documents, there is little optioning available for deducing granular corpuses based on these entity labels.



At this time, no NER was utilized to find set overlap within the body of the text as this would open a host of additional challenges with coreference resolution. The labels already tagged within the MIND dataset had aliases attached to the entity that were taken into consideration.

Instead, groupings were developed empirically to gradually increase the scale of the corpus being evaluated. These corpuses were derived from key word searches throughout the text of the documents being processed. The groupings were finalized as:

- University of Connecticut Women's Basketball Team (3 documents)
- Kansas University (5 documents)
- Basketballs Games in general (116 documents)
- Zion Williamson (145 documents)
- Basketball in general (4880 documents)
- Sports in general (31106 documents)

The progressive size of the corpuses – scaling with the broadness of the category – allows us to look at accuracy and feasibility of summarizing a corpus of that size.

Running OCCAMS Experiment w/ Defined Corpus

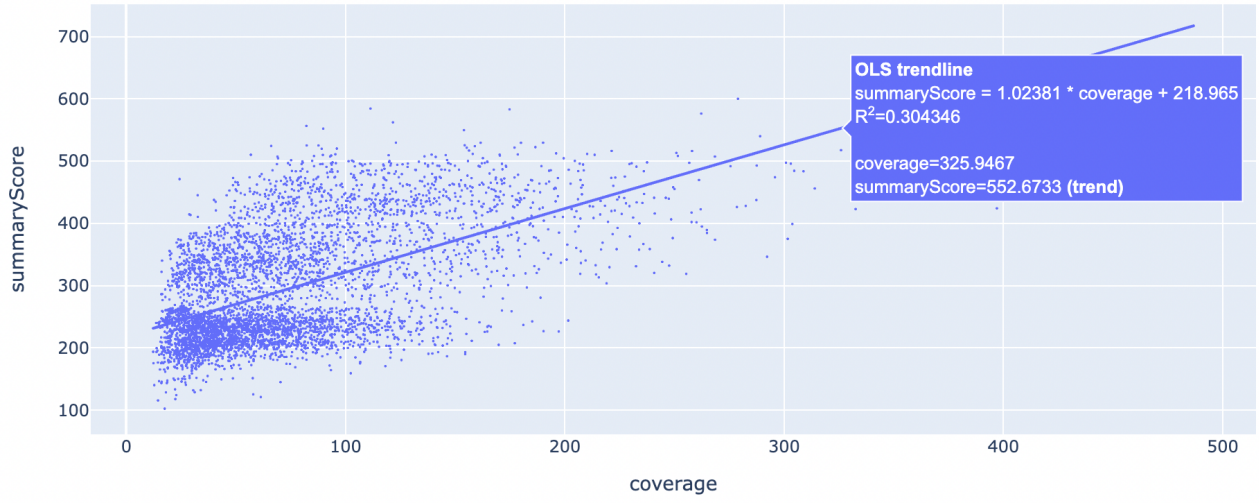
Once these corpuses were identified, they were each used in testing against each summary scheme utilizing 3 different target length (100/150/250 words). The results of the experiment output can be found in the repo folder 'out/experiment.xlsx'. The defined above metrics were utilized to do some preliminary EDA regarding if any of the hyperparameters chosen could be utilized in a robust summarization tool for analysts. Again, information synthesis (coverage) and accuracy were the two main attributes that we need to prioritize when considering analysts as a customer.

Additional information on running the experiment (or individual corpus summarization) can be found in EXPERIMENT.md

OCCAMS RESULTS

Information Gain

OCCAMS provides a *summaryScore* that is a score of relative goodness for summarization that is calculated for each sentence in a corpus. The OCCAMS *summaryScore* was evaluated against several of the metrics established. The only comparison which was positively correlated was *summaryScore* and coverage.



However, this coverage wasn't equally dispersed amongst each document in the corpus. When looking at the average number of entities utilized from each document in compiling the summary, a bias was found toward the document with the terms having the highest weight. This 'primary' document in the corpus held most of the unique entities in the summary; the other documents in the corpus then mimicked this main document to include duplicative entities, rather than new entities (indicative of new information). In this sense, adding a hyperparameter to decrease the number of entities overlap a

sentence could contain may increase information gain; albeit, the opportunity for presenting a summary with less value arises when terms with lower weight are being brought in.

Generalizing, utilizing a metric of coverage would be an approximate for OCCAMS summary score, with the intent that higher unique entity presence in summaries lead to more information gain.

Highlights Rather than Summaries

Although we address OCCAMS as a summarizer, the sentences are simply extracted from the text of the corpus. While this leads to factual statements – although potentially misleading if the context is not also extracted – the readability of results was poor. Rather, adapting the term highlights tended to be a more accurate statement for the state of the summaries. Highlights were then presented in a user-friendly bulleted list, without trying to obfuscate that these sentences are unrelated. For intelligence analysts use, this method of summarization lacked coherency and – due to the sometimes-missing context for the extracted sentences – many highlights needed to be disregarded.

Abstractive summarization methods would certainly support coherency, but hallucinations are hypothesized to make this impractical for intelligence use cases. Potential opportunities moving forward include:

- Producing summarizations using abstracted methods with a preference for utilizing direct text from primary sources. Coherency would improve and the risk of hallucinations would be reduced.
- Utilizing a text generator to reword text from an extractive summarizer. The goal being that no additional knowledge is being added to the summary, nor is the model asked to prioritize the value of information. Rather, the text is rewriting a subset of the article selected by the extractive summarizer.

Further testing would be needed with both proposed methods to see if the information gain would be valuable.

Cross Referencing with an Existing Knowledge Base

Another direction – which currently is not integrated into OCCAMS – is to use an existing knowledge base as a way of solidifying which information is of interest. Requests For Information (RFIs) could be thrown at articles, with high scoring answers dictating which sentences to be prioritized for extraction. These prioritized sentences in turn could bias the term weights toward information relevant to support these facts. While the Fisher summarization scheme attempts to do this by biasing term weights to a particular background corpus, summarization results become heavily influenced by similar writing styles as opposed to content.

Adapting Experiment to External Summarization Models for Comparison

For comparison, we have run the same article against OCCAMS and ChatGPT. Here are the results:

	OCCAMS	ChatGPT
method	statistical budget-based optimization problem	transformer tuned on internet text through 2021
short_summary	If the UConn women's basketball team is going to be good this season — as in, back-to-the-Final Four good — Crystal Dangerfield will have to be great. "And [the point guard] could score. Sophomore Christyn Williams is one of the most talented players in the sport and that she ended up on the preseason second team, and not the first, is a joke. How many of those did you have as a sophomore? You know what the chances are of the Huskies advancing for the 13th year in a row without a healthy and consistent Dangerfield taking over and pulling? Crystal is going to have to do that every game.	"The article discusses how the success of the UConn women's basketball team this season will depend heavily on the performance of senior point guard Crystal Dangerfield. With several key players having left, Dangerfield is the only senior on the team and will be expected to lead. Coach Geno Auriemma noted that the team doesn't have many "ready-made giants of the sport" for Dangerfield to pass to, so she will have to score more than some of the program's previous point guards. Although Dangerfield has shown flashes of greatness, she has also faded at times, so she will need to perform consistently at a high level for the team to be successful."
long_summary (200 words)	If the UConn women's basketball team is going to be good this season — as in, back-to-the-Final Four good — Crystal Dangerfield will have to be great. "And [the point guard] could score. That makes her job a little more difficult. Junior Megan Walker, named all-conference first team, is capable of a breakout season. Sophomore Christyn Williams is one of the most talented players in the sport and that she ended up on the preseason second team, and not the first, is a joke. How many of those did you have as a sophomore? Last year's UCLA game, where she just took over and almost singlehandedly won the game, how many did you have? Of leading the team, Dangerfield said, "You have to take more responsibility for what happens, what goes wrong. Crystal is going to have to do that every game. UConn women's basketball fans got their first look at the 2019-20 Huskies, including three of the program's newcomers, in a 103-40 exhibition win over Division II Jefferson Sunday afternoon at Gampel Pavilion. Evelyn Adebayo was the last available player to come off the bench with under two minutes remaining in the second half.	The success of the UConn women's basketball team this season is dependent on the performance of their senior point guard, Crystal Dangerfield. She is the team's only senior player and has been named the American Athletic Conference's preseason co-player of the year. Dangerfield is expected to play a crucial role in a season that could either be a continuation of the team's winning streak or a period of transition. With key players Napheesa Collier and Katie Lou Samuelson no longer on the team, UConn has a relatively inexperienced roster, and it's uncertain how they will fare in the upcoming season. Dangerfield will have to take on a leadership role and score more to compensate for the absence of All-American players. She had surgery to correct a hip impingement during the off-season but is expected to be fully recovered. Head coach Geno Auriemma hopes that Dangerfield will be consistently great this season, citing her performance in the past when she scored 19 points, five assists, and four rebounds in a game against Baylor during her freshman year, and when she helped the team survive a Sweet 16 scare against UCLA last season. The Huskies have the potential to win a national championship, but it will depend on Dangerfield's performance and the development of the rest of the team.

Overall, the coherency of the NN is far superior. Manually fact checking, both summarizers end up factually correct, and ChatGPT even did abstract some direct quotes from the articles to make the argument more convincing. Further testing would be needed to address concerns of hallucinations.

MOVING FORWARD

In order to ensure more success is found with summarization, emphasis should be placed on defining what a ‘unique’ and ‘relevant’ piece of information is. Primitive heuristics that define relevance based on attributes within a corpus have not been deemed to be a comprehensive enough subset of data to produce quality summarizations. Current summarizations read more as ‘highlights’ of a document and do not produce high level comprehension across a given corpus. Rather, summarization models could be adapted to integrate corpuses with existing databases of knowledge that can be used in comparison for relevant and new pieces of information. Attributes extracted from the corpus itself has proved insufficient to produce valuable summaries. In addition, in the case of multi-document summarization, clustering techniques would benefit the aggregation of information across documents to ensure that the summarizations recognized trends across documents.