‘Keep it Together’: Enforcing Cohesion in Extractive Summaries by Simulating Human Memory

Anonymous ACL submission

Abstract

Extractive summaries are usually presented as lists of sentences with no expected cohesion between them. In this paper, we propose a method to enforce cohesion whilst controlling for redundancy in summaries, in cases where the input exhibits high redundancy. The pipeline controls for content redundancy in the input as it is consumed, and balances informativeness and cohesion during sentence selection. Our sentence selector simulates human memory to keep track of cohesive chains while building the summary, enforcing cohesive ties between noun phrases. Extensive experiments, both automatic and human, revealed that it is possible to extract highly cohesive summaries that are as informative as summaries optimizing only for informativeness. The extracted summaries exhibit a smooth topic transition between sentences as signaled by lexical chains, with chains spanning adjacent or near-adjacent sentences.

1 Introduction

Automatic text summarization is the task of processing a document(s) and producing a shorter text, the *summary*, that retains the gist of the information, with many variations along the years (Nenkova et al., 2011). *Extractive* summarization selects content units (usually sentences) and presents their concatenation as the summary. It remains challenging to control the specific content units so that the summary ends up being non-redundant and informative, with much previous work modeling these qualities during document understanding (Peyrard et al., 2017; Xiao and Carenini, 2020; Gu et al., 2022). However, coherence control has received less attention (Barzilay and Lapata, 2008; Wu and Hu, 2018), partly because merely reliably evaluating whether a text is coherent remains challenging (Goyal et al., 2022; Steen and Markert, 2022; Zhao et al., 2023).

We introduce an extractive summarization methodology that implements two control mechanisms at different stages of processing: the first one to control redundancy during input understanding, and the second one to control the trade-off between informativeness and cohesion during summary extraction. Cohesion is the property of a text to function as a unified whole, exhibiting thematic links –called *cohesive ties*– between nearby sentences (Hassan and Halliday, 1976). In contrast, coherence refers to the discourse organization of a text, usually signaled by discourse markers. When building extractive summaries by concatenating sentences, we argue that controlling for cohesion is a better-defined task than aiming to control coherence, especially if no sort of post-editing (e.g. replacing discourse markers) is applied (Zajic et al., 2007; West et al., 2019; Mallinson et al., 2020). A potential benefit of producing a more cohesive text is that it is easier to read and understand for humans, especially when the knowledge domain is highly technical, as reported by previous work in psycholinguistics (Kintsch, 1990) and automatic summarization (Barzilay and Elhadad, 2002).

In our pipeline, summary properties are controlled in the following way. On the one hand, summary redundancy is addressed by controlling the redundancy levels of the input text, following previous findings (Carbonell and Goldstein, 1998; Xiao and Carenini, 2020). The pipeline consumes input text in a cascaded way: first splitting the input into contiguous passages, then consuming passages one at a time so as to minimize their semantic similarity with already selected passages.

On the other hand, informativeness and cohesion are directly modeled during summary extraction. Extraction is done in a sentence-by-sentence fashion, quantifying summary properties independently at each step. The objective is to select a highly cohesive sentence that is informative enough. We introduce a sentence selector that incrementally builds cohesive chains of noun phrases and models chain interaction. The selector, KvD-SELECT, keeps track of chains currently active by simulating human memory according to the Micro-Macro theory, henceforth KvD (Kintsch and van Dijk, 1978), a psycholinguistic theory of discourse comprehension and production. Working memory –a type of short-term memory– is modeled as a limited-capacity buffer of lexical chains, forcing the model to keep only the most salient chains.

We test our methodology on newswire multi-document summarization and single-long document summarization of scientific articles, patents, and government reports. Across domains, extensive experiments show that, first, our system is effective at incrementally building an input sequence with lower content redundancy, which translated to a significant reduction in summary redundancy. Second, the proposed sentence selector managed to maintain summaries informative while improving cohesion significantly: over 15% more noun phrases and over 20% more sentences were connected through cohesive ties w.r.t a greedy selector. Tailored human evaluation campaigns revealed that cohesion has a positive impact on perceived informativeness, and that our extracted summaries exhibit chains covering adjacent or near-adjacent sentences. Closer inspection showed that topics flow smoothly across extracted summaries with no abrupt change or jumps.

In summary, our contributions are as follows:

- We propose a cascaded encoder capable of consuming arbitrary long textual input that controls the level of content redundancy the rest of the pipeline is exposed to.
- We propose a summary extraction method that models informativeness and cohesion independently and allows to control the balance between the two when building the summary.
- Automatic and human experiments show the effectiveness of our control mechanisms and how summary properties can be balanced according to user needs in a straightforward way.

2 Related Work

Previous work has modeled cohesion during document understanding by keeping track of tied named entities (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013), topic flow (Barzilay and Elhadad, 2002), or by implementing discourse theories (Jeon and Strube, 2020). Most similar to our approach, Fang (2019) introduced an implementation of the KvD theory that models cohesion and informativeness during document understanding, assigns a single importance score to each sentence, and employs a greedy sentence selector. In contrast, we quantify summary properties separately, and model cohesion by implementing KvD during sentence selection. This approach provides a more explicit way to control the contribution of each property during summary extraction.

Similar ways to control summary properties during summary selection have only focused on minimizing redundancy (Carbonell and Goldstein, 1998; Fabbri et al., 2019; Xiao and Carenini, 2020), where the extractive summary is regarded as a list of sentences with no particular order to them, a design choice possibly influenced by the format of available benchmarks such as CNN/DM (Hermann et al., 2015) and DUC. However, seminal work highlighted the role of redundancy in text (Walker, 1993; Tauste, 1995), and how its presence is a result of human memory limitations (Johnstone, 1994).

In this work, we provide evidence that controlling for cohesion constitutes a better strategy for providing the end-user with a more comprehensible summary, formatted as a multi-sentence cohesive text instead of a list of sentences. Our results show that this setup is especially effective when the knowledge domain is highly technical, and when a sentence ordering cannot be inferred from the input trivially, e.g. in multi-document summarization.

3 Problem Formulation

We tackle the task of extractive summarization as a sentence scoring step followed by a selection step. Figure 1 shows the pipeline of the system, in which sentences are scored in a cascaded fashion, as follows. First, the input is segmented into blocks of contiguous sentences and the block selector module then selects blocks based on their relevancy and their redundancy w.r.t. already selected blocks. Second, a local encoder obtains block-level representations for each sentence in the block. After all document blocks are processed, all these encodings are concatenated into a single embedding sequence and passed to the global context encoder, which will obtain a document-aware representation of each sentence. Finally, a selection module will extract a subset of sentences and present them as the summary in the order they were extracted. The pipeline is designed to be capable of consuming documents of arbitrary length, offering further control over levels of information redundancy the

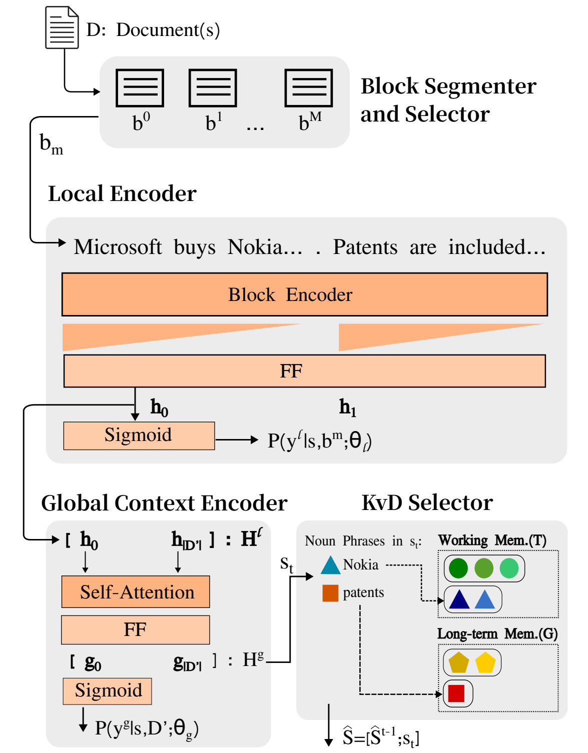


Figure 1: Our extraction pipeline: local extraction step m adds local sentences to D'; at sentence selection step t, KvD-Select balances informativeness of candidate s_t with cohesion of summary $\hat{S}$.

sentence selector is exposed to. We now proceed to elaborate on each module of the proposed pipeline.

3.1 Block Segmentation and Selection

Processing starts by segmenting the document(s) D into fixed-length overlapping blocks, each of which includes preceding and subsequent wordpieces, providing surrounding context. Then, blocks are selected iteratively until a predefined budget (total number of wordpieces) is met. At step m, block b_m is selected such that

$$b_m = \operatorname*{argmax}_{b\in B\setminus\hat{B}}[\lambda_b \mathrm{LR}(b) - (1-\lambda_b)\max_{b_j\in\hat{B}} \mathrm{Sim}(b,b_j)] \quad (1)$$

where $\hat{B}$ is the set of blocks already selected, $\mathrm{Sim}(x,y)$ is the cosine similarity between TF-IDF vectors of blocks x and y, and λ_b allows to control the mix of both terms. $\mathrm{LR}(b)$ is the continuous LexRank score of block b (Erkan and Radev, 2004), calculated over the complete graph of blocks in D,

$$\mathrm{LR}(b) = \frac{d}{|B|} + (1-d)\sum_{v\in \mathrm{adj}[b]} \frac{\mathrm{Sim}(b,v)}{\sum_{z\in\mathrm{adj}[v]}\mathrm{Sim}(z,v)}\mathrm{LR}(v)$$

where d is the damping factor and $\mathrm{adj}(b)$ is the set of block nodes adjacent to b. This module balances block relevancy (as proxied by centrality) and input redundancy in a straightforward way by linearly combining their scores. After an optimal block is selected, it is sent to the local encoder module.

3.2 Local Encoder (LE)

Given block b as a sequence of wordpieces spanning contiguous sentences, the local encoder will obtain representations for each sentence covered in b. This module is trained as a local extractive summarizer itself, under sequence labeling formulation where each sentence in the block is labeled as $y_i^\ell \in \{0,1\}$ to indicate whether sentence s_i is selected or not. Then, sentence representation h_i is defined as the average embedding over s_i wordpieces, obtained from a LongT5 encoder (Guo et al., 2022). Finally, the probability of s_i being locally selected is defined as $P(y_i^\ell \mid s_i, b; \theta_\ell) = \sigma(W^\ell \cdot h_i)$, and the module is trained using cross-entropy loss independently from the rest of the pipeline. During inference, the local encoder consumes one block, selects N sentences and adds them to D' –containing all locally selected sentences so far–, and their corresponding embeddings to H^ℓ.

3.3 Global Context Encoder (GCE)

Given the sequence of local sentence embeddings H^ℓ, this module obtains the sequence of globally-aware representations H^g as follows. Sequence H^ℓ is passed through a self-attention layer (Vaswani et al., 2017), i.e. $g_t = \mathrm{SelfAttn}(h_t, H^\ell), \forall h_t \in H^\ell$. Similarly to the LE module, the probability of selecting $s_t \in D'$ is $P(y_t^g \mid s_t, D'; \theta_g) = \sigma(W^g \cdot g_t)$, where $y_t^g \in \{0,1\}$ indicates whether s_t is selected or not for the final, global summary, and also trained using cross-entropy loss.

3.4 Sentence Selection

Finally, candidate summary $\hat{S}$ is built by selecting one sentence at a time from D', taking into account the informativeness and cohesion of each candidate sentence w.r.t. the already selected sentences. At selection step t, the optimal sentence is given by

$$s_t = \operatorname*{argmax}_{s\in D'\setminus \hat{S}^{t-1}} [\lambda_{\mathrm{sel}} f_{\mathrm{I}}(s) + (1-\lambda_{\mathrm{sel}}) f_{\mathrm{C}}(\hat{S}^t)] \quad (2)$$

where function f_{I} estimates the informativeness of candidate sentence s, f_{C} estimates the cohesion of candidate summary $\hat{S}^t = [\hat{S}^{t-1}; s]$, and $\lambda_{\mathrm{sel}} \in [0,1]$ is a parameter that allows to control their trade-off. Following Xiao and Carenini (2020), we take the probability of selecting s given by the global context encoder module as a proxy for informativeness, i.e. $f_{\mathrm{I}}(s) = P(y^g \mid s, D'; \theta_g)$. In the next section, we elaborate on how f_{C} models and enforces cohesion during sentence selection.

4 Cohesion during Summary Extraction

Cohesion is a language mechanism that enables a sequence of sentences to function as a unified whole (Hassan and Halliday, 1976). It does so by linking semantic units in a text through *cohesive ties*, regardless of the grammatical or discourse structure these units are part of. In particular, lexical cohesion links units with the same lexical form, synonyms, or units in the same semantic field. Furthermore, units tied cohesively can be grouped in chains by their semantic similarity. Whilst the mere presence of two or more chains does not guarantee a cohesive effect, their interaction can be a reliable proxy for cohesion (Morris and Hirst, 1991; Barzilay and Elhadad, 1997).

In this paper, we focus on modeling lexical cohesive ties between noun phrases in nearby sentences of a summary by controlling the interaction between lexical chains.

4.1 KvD Select

The proposed selector, KVD-SELECT, calculates cohesion score f_C by simulating the processes in working memory during text production. The procedure is based on the Micro-Macro Structure theory (Kintsch and van Dijk, 1978), which describes the cognitive processes involved in text comprehension and production at the local (micro) and global (macro) level of discourse. Following Fang (2019), we implement processes happening at micro-level, which deal with the movement of content in and out of working memory.

Let T be working memory and G long-term memory (LTM), where both are separate sets of cohesive chains, and each chain as a set of noun phrases (NPs). At selection step t, the algorithm extracts NPs from s_t and connects them to the chains in T and G, constraining the number of active chains in T afterward. Cohesive score f_{C} then depends on the average similarity between units added to T and those added to G. We now elaborate on each step of the algorithm.

Extracting Noun Phrases. Given sentence $s_t \in D'$, we obtain P, the set of extracted nominal chunks, obtained by merging nominal nodes in dependency trees with their children, following the procedure of Fang (2019). Specifically, given that node *u* is nominal dependent of a clausal predicate, *u* will have its child *v* merged if either*v* is a function word, a single-token modifier, or *u* and *v* form part of a multi-word expression.

Adding Content to Memory. Next, cohesive ties between s_t and $\hat{S}^{t-1}$ are enforced by adding each NP in P to the chain with the highest element-wise semantic similarity. Formally, the optimal chain to add $a \in P$ to is $C^* = \mathrm{argmax}_{C \in T}\{\phi(p, C)\}$, where ϕ is the average BERTScore (Zhang et al., 2019) between a and each NP in C. In order to make sure that chains maintain an acceptable level of semantic similarity between elements, a is added to chain C only if $\phi(a, C) \geq \nu$, where ν is the minimum admissible similarity. This way the algorithm can control the similarity length between chain members, and avoid a single, long chain.

If similarity with chains in T is not strong enough, we look at chains in G, in which case the chosen chain is moved back to T. This step simulates how humans recall content no longer present in WM, the *recall mechanism* (Kintsch and van Dijk, 1978). If still no chain in G meets the similarity requirement, we proceed to create a brand new chain in T with a as its sole element. By searching for a good enough candidate chain first in T and then in G, we encourage cohesive ties between NPs in nearby sentences.

Updating Memory. After adding incoming NPs to chains in memory, T is updated to retain only the `WM` most recent chains, where *recency* of a chain is defined as the id of the selection step in which this chain was last retained in T. For instance, a chain currently in T is more recent (higher step id) than a chain in G discarded in an earlier step. This design choice mimics the *recency effect* behaviour during *free recall* tasks in human subjects (Glanzer, 1972), a behaviour attributed to short-term memory. Finally, discarded chains are moved to G, concluding the selection step.

Candidate Scoring. Next, we define cohesion score f_{coh} which will be used to discriminate amongst possible continuations to $\hat{S}^{t-1}$. The objective is to encourage NPs in P to be assigned to recent chains, in turn encouraging chains to cover nearby sentences in the final summary. In addition, we want to score down candidate sentences with NPs added to chains in long-term memory.

Let $A_T = \{a; a \in P, C_a \in T\}$, where C_a is the chain a was added to. Similarly, let $A_G = \{b; b \in$

$P, C_b \in G\}$. Then, let rec(C) be the number of selection steps passed since the last time chain C was retained in T. Quantity rec(C) functions as a proxy for how spread chain C is, i.e. how far away two sentences covered by C are. Then,

$$f_{\text{coh}} = \frac{1}{|A_T|} \sum_{a \in A_T} \frac{\phi(a, C_a)}{\text{rec}(C_a)} + \frac{\gamma_{\text{rec}}}{|A_G|} \sum_{b \in A_G} \frac{\phi(b, C_b)}{\text{rec}(C_b)}.$$

Hence, the cohesive score depends on the contribution of each cohesive tie formed. For each chunk in A_T and A_G, its contribution depends directly on the strength of similarity to its assigned chain and inversely on the spread of said chain. The contribution of chunks in A_G is scaled down by hyper-parameter $\gamma_{\text{rec}} \in [0; 1]$ as to simulate the higher cognitive cost incurred when retrieving information from long-term memory.

5 Experimental Setup

We now describe the datasets used, training details, baselines, and evaluation methodology.

5.1 Datasets

We consider datasets for single-document summarization of long, highly redundant documents, and multi-document summarization:

- **PubMed.** Scientific articles in the biomedical domain collected from PubMed (Cohan et al., 2018). We use text from all sections as the source document and the abstract as reference summary.
- **BigPatent.C.** Patents in the Chemistry and Metallurgy industry (Sharma et al., 2019).
- **GovReport.** Long legislature reports (Huang et al., 2021) of U.S. bills.
- **MultiNews.** Collections of news articles paired with reference summaries (Fabbri et al., 2019).

5.2 Pipeline Parameters

Hyper-parameters were tuned over the validation sets of each dataset.

Document Segmentation and Block Selection. We use a block size of $B = 2048$, context size of $C = 200$ pieces, $\lambda_b = 0.2$, and set a budget of 16 384 input wordpieces.

Local Encoder (LE), Global Context Encoder (GCE). The block encoder in LE is initialized with a pretrained checkpoint of LongT5 with transient-global attention (Guo et al., 2022),[1] and an output layer of size 200.

The LE module is trained independently from the GCE module, with LE being trained first, then GCE trained whilst LE remains frozen. In both cases, we used the Adam optimizer (Loshchilov and Hutter, 2018), a constant learning rate of $1e^{-6}$, effective batch size of 64, and 50k training steps.

For training LE, we obtain extractive oracle sentences from each block and train the module over blocks with ROUGE-1 + ROUGE-2 > 0.5. During inference, we extract a maximum of $N = 10$ local sentences per block and a maximum of 1000 sentences in total.

Summary Extractor. We set $\lambda_{sel} = 0.8$, working memory `WM`$= 6$, recall cost $\gamma_{\text{rec}} = 0.01$, and a minimum NP similarity of $\nu = 0.6$. Word budget is set to 200, 100, 650, 250 for PubMed, BigPatent.C, GovReport, MultiNews, respectively.

5.3 Comparison Systems

We compare against the standard extractive oracle, EXT-ORACLE, obtained by greedily selecting sentences maximizing ROUGE-1 + ROUGE-2 against gold summaries until the word budget is met. For cohesion analysis, we also report metric values over the gold summaries, labeled as GOLD.

The impact of cohesion modeling is assessed by employing a greedy selector over GCE scores, equivalent to set $f_C = 0$ in Eq. 2, dubbed LT5-CASC. In addition, we report LongT5 performance when consuming the input as a flat sequence and using a greedy selector, dubbed LT5-FLAT.

Regarding alternative sentence selectors, we compare against the following.

MMR-Select. (Xiao and Carenini, 2020) Reduces redundancy by selecting s_i (candidate sentence at selection step i) such that cosine similarity w.r.t. the partially extracted summary $\hat{S}$ is minimized. Informativeness and redundancy are balanced in the same way as in Eq. 2.

N-gram passing (NPassing). Encourages repetition ties by allowing p percent of n-grams in s_i to overlap with $\hat{S}$. When $p = 0$, this method reduces to n-gram blocking, whereas when $p = 1.0$, to greedy selection. We report bi-gram passing.

Semantic Similarity Distribution (KL-Dist). Models the intuition that noun phrases in s_i will be

[1] HuggingFace, `google/long-t5-tglobal-base`

more semantically similar to some units in $\hat{S}$ whilst dissimilar to others (Taboada, 2004). Let $\hat{Q}_i$ be the similarity distribution obtained when comparing every NP in s_i against every NP in $\hat{S}$. Similarly, let Q be the distribution of similarity between NPs in different sentences in gold summaries. Then, $f_C = \exp\big(-D_{KL}(Q||\hat{Q}_i)\big) - 1$, where D_{KL} is the Kullback–Leibler divergence. Higher values of f_C indicate lower diverge, encouraging $\hat{S}$ to have a cosine similarity distribution similar to those seen in gold summaries. All distributions were discretized into 20 bins covering values from -1.0 to 1.0.

Shuffle Classifier (CCL-Select). Holistically quantifies coherence using CCL (Steen and Markert, 2022), a scorer trained to distinguish shuffled from unshuffled text that showed a high correlation with human ratings of coherence. We use RoBERTa (Liu et al., 2019) as underlying model and use a window of 3 consecutive sentences.

5.4 Evaluation

Informativeness is assessed using ROUGE F_1 (Lin, 2004). Redundancy is evaluated using sentence-wise ROUGE-L (Bommasani and Cardie, 2020), dubbed *RdRL*. Additionally, we define *Inverse Uniqueness (IUniq)*, $1 -$ Uniqueness, where 'Uniqueness' (Peyrard et al., 2017) is the ratio of unique n-grams to the total number of n-grams. We report the mean value between uni-, bi-, and trigrams. Higher values denote higher redundancy.

Cohesion is evaluated with the followed metrics: *CoRL*, the average ROUGE-L F_1 between consecutive sentences; and *Entity Graph (EGr)* (Guinaudeau and Strube, 2013), which models a text as a sentence graph with edges between sentences with nouns in common, using the average edge weight as a proxy for cohesion. Finally, coherence is assessed using CCL (Steen and Markert, 2022).

5.4.1 Human Evaluation

We elicit human judgments to assess overall quality, informativeness, and cohesion in two separate studies. We sampled 30 documents from the test set of PUBMED and compare systems LT5-CASC, MMR-SELECT, and KVD-SELECT.

Ranking Campaign. Subjects were shown the abstract and the introduction of a scientific article along with two system summaries, and then then asked to select the best summary (or select both in case of tie) according to three criteria: (i) overall quality, (ii) informativeness, and (iii) cohesion. In this setup, cohesion is evaluated as a holistic property of the text, as perceived by a reader.

Chaining Campaign. Subjects were shown a single summary and were asked to annotate lexical chains by grouping together pre-extracted NPs in the same semantic field. We report the following chain properties: (i) *chain spread*, defined as the average number of sentences between immediate-neighbor sentences covered by the same chain;(ii) *chain density*, the number of chains covering the same sentence; and (iii) *sentence coverage*, the percentage of sentences covered by at least one chain.

Inter-annotator agreement is calculated as the average lexical overlap between chains, expressed in F_1 score, calculated pair-wise between subjects. For this study, we include reference summaries as one more analysis system.

6 Results and Discussion

Next, we discuss the results of our analyses and the outcome of the human evaluation campaigns.

6.1 Reducing Redundancy in Input Blocks

The following block selection strategies were compared: (i) *Original*, consisting of selecting blocks in their original order in the source document;[2] (ii) *Oracle Selection*, which selects the block that maximizes ROUGE F_1 scores (mean of ROUGE-1 and ROUGE-2) w.r.t. the reference summary; (iii) *Max. Redundancy*, which selects the most similar block possible (by flipping the sign in Eq. 1); and finally, (iv) BLOCKSELECT, the proposed strategy.

The analysis, showcased in Figure 2, evaluates input redundancy at each block selection step, as well as informativeness and redundancy of summaries extracted from the blocks available at each step, using a greedy selector. The results indicate that the strategy used to select input blocks has a direct impact not only on input redundancy –as intended– but also on summary redundancy.

Notably, BLOCKSELECT is effective at incrementally building an input sequence with lower content redundancy. Compared to the other strategies, ours has a clear impact on summary redundancy, enabling the pipeline to consistently extract summaries that are significantly less redundant. Similar trends were observed in the other datasets.[3]

[2] For multi-document datasets, we use the order provided in the dataset release.

[3] See Fig. 3 in Appendix C.3 for results in other datasets.

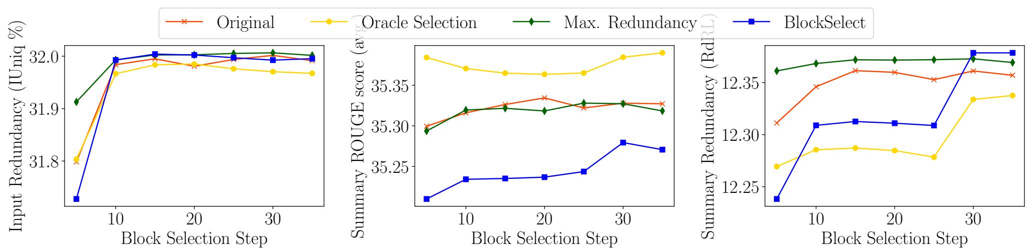


Figure 2: Effect of block selection strategy over input redundancy (left), summary informativeness (mid), and summary redundancy (right), evaluated as block selection proceeds on the MULTINEWS validation dataset.

6.2 Trading-off Informativeness and Cohesion

Next, we turn to the summary extraction module. Tables 1 and 2 present the performance of all compared system in terms of informativeness and cohesion, respectively. In all our experiments, statistical significance at the 95% confidence level is estimated using Mann–Whitney U tests ($p < 0.05$).

First, note the impact on cohesion when controlling for redundancy. MMR-SELECT indeed manages to obtain comparable informativeness levels to LT5-CASC, being most effective for BIGPATENT.C. However, minimizing sentence similarity comes at the expense of a significant decrease in cohesion (CoRL) and coherence (CCL). Second, we find that NPASSING is the only one capable of obtaining comparable or better ROUGE scores but CoRL and EGr scores indicate that lexical passing is not enough to improve cohesion. Next, note that KL-DIST employs a seemingly more aggressive trade-off between ROUGE and CoRL in all datasets except PUBMED. We hypothesize that its cohesion term, f_C, saturates the final candidate score during trade-off, which prompts the selector to pick candidates with lower informative scores.

When guiding selection with a holistic shuffle scorer, as expected, CCL-SELECT obtains remarkably high CCL scores, closing the gap w.r.t. EXT-ORACLE in most datasets and even surpassing it for BIGPATENT.C. However, note that this selector does show a significant reduction in CoRL and EGr scores w.r.t. LT5-CASC, indicating that CCL is measuring also discourse organization, possibly in the form of rhetorical role ordering –first background, then method, and so on. Hence, it can be said that summaries in CCL-SELECT are better organized in terms of rhetorical roles but exhibit lower cohesion than greedily selected summaries.

Finally, KVD-SELECT manages to strike an even more aggressive trade-off between informativeness and cohesion. Across datasets, the selector exhibits lower ROUGE scores but the best CoRL, EGr scores (except for PUBMED), and second highest CCL score after CCL-SELECT.

Effect of Parameter λ_{sel}. Next, we analyze how summary properties vary across increasing levels of λ_{sel}. Note that when $\lambda_{\text{sel}} = 0$ selectors depend entirely on f_C, and $\lambda_{\text{sel}} = 1.0$ is the greedy selector.

As expected, we found that informativeness is higher as f_I is weighted up (higher λ_{sel}) with all selectors except MMR-SELECT. This indicates that it is possible to increase cohesion without incurring a significant loss in informativeness. Interestingly, KVD-SELECT seems robust to λ_{sel} in terms of CoRL and RdRL. We hypothesize that KVD-SELECT benefits from a signal indicating which cohesive ties are informative and worth enforcing.

Impact of Cascaded Processing. When comparing systems that used flat input vs cascaded input (LT5-FLAT and LT5-CASC), we found that cascaded processing exhibits lower ROUGE scores than flat processing in PUBMED and MULTINEWS, and comparable performance for BIGPATENT.C and GOVREPORT. However, LT5-CASC shows slightly higher CoRL scores in all datasets. This indicates that cascaded processing puts a greedy selector in a better position to extract more cohesive summaries, however at the expense of a slight decrease in informativeness.

6.3 Human Evaluation

In both studies, statistical significance between system scores was assessed using a one-way ANOVA with posthoc Tukey tests with 95% confidence interval ($p < 0.01$). Results are presented in Table 3.

Ranking. Krippendorff's α (Krippendorff, 2011) showed an inter-annotator agreement of 0.68. For overall quality, subjects showed a significant preference for KVD-SELECT over LT5-CASC. For cohesion, KVD-SELECT was perceived as more

System	PubMed			BigPatent.C			GovReport			MultiNews		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Ext-Oracle	65.10	37.99	60.76	53.85	23.20	46.90	72.66	40.90	69.36	62.66	33.73	57.93
LT5-Flat	**48.15**†	**21.45**†	**44.49**†	39.54	13.25	**34.30**	59.33	25.94	56.29	**47.07**	**17.54**†	**42.96**
LT5-Casc	46.16	19.74	42.49	39.57	13.25	34.26	59.73	26.21	56.50	46.80	17.21	42.66
+MMR-Select	46.14	19.63	42.47	**39.59**	**13.29**	**34.30**	**59.79**	**26.30**	**56.56**	46.76	17.13	42.59
+NPassing	46.38	19.92	42.74	**39.59**	13.26	34.29	**59.79**	26.25	**56.56**	46.91	17.27	42.78
+KL-Dist	46.00	19.62	42.32	39.25	13.07	33.89	59.46	25.85	56.15	46.63	16.97	42.45
+CCL-Select	45.91	19.60	42.45	39.16	12.95	33.92	59.72	26.24	56.50	46.85	17.29	42.71
+KvD-Select	44.90†	18.47†	41.27†	38.37†	12.41†	33.13†	57.88†	23.66†	54.57†	45.85†	16.13†	41.62†

Table 1: Summary informativeness in terms of ROUGE scores (R1, R2, RL). †: Scores are statistically different from the closest system. Best systems are **bolded**; systems better than LT5-CASC shown in blue and worse, in red.

Systems	PubMed			BigPatent.C			GovReport			MultiNews		
	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL
Gold	14.45	0.95	0.78	19.19	0.78	0.83	16.21	1.95	0.75	10.45	0.71	0.80
Ext-Oracle	14.68	0.99	0.42	15.94	0.68	0.41	16.55	1.92	0.51	10.85	0.68	0.50
LT5-Flat	16.60	**1.10**	0.26	19.76	0.75	0.37	16.06	2.00	0.28	12.25	0.91	0.26
LT5-Casc	**17.47**	1.07	0.26	20.26	0.73	0.39	16.46	2.04	0.27	12.51	0.90	0.26
+MMR-Select	16.89†	1.07	0.25†	18.82†	0.73	0.37	15.88†	2.03	0.27	11.83†	0.88	0.25†
+NPassing	16.66†	1.07	0.27	19.91	0.73	0.39	16.38	2.04	0.27	12.17	0.89	0.26
+KL-Dist	17.31	1.08	0.27	20.54	0.73	0.41	16.88†	2.05	0.27	12.82†	0.95†	0.26
+CCL-Select	17.28	1.06†	**0.48**†	19.41†	0.71†	**0.66**†	16.73	2.04	**0.45**†	11.94†	0.86†	**0.46**†
+KvD-Select	17.32	1.05†	0.28†	**22.21**†	**0.78**†	0.42	**18.88**†	**2.15**†	0.29†	**14.23**†	**0.99**†	0.29†

Table 2: Summary cohesion in terms of consecutive ROUGE-L score (CoRL) and EntityGraph (EGr), as well as coherence (CCL). For all metrics, higher is better. See Table 1 for formatting details.

cohesive compared to LT5-CASC, and LT5-CASC was more cohesive than MMR-SELECT.

Chaining. Chain overlap was calculated at 0.90. Differences between LT5-CASC and all other systems, as well as MMR-SELECT–GOLD and KVD-SELECT–LT5-CASC were found to be significant, for all measurements of cohesion. Moreover, the number of NPs annotated per chain was 2.30, 2.33, 2.80, and 2.55, for systems LT5-CASC, MMR-SELECT, KVD-SELECT, and GOLD, respectively.

We found that KVD-SELECT summaries exhibit more active and denser chains and better-covered sentences than the baselines. Note that LT5-CASC obtains the lowest chain spread but also low coverage, indicating that its summaries exhibit very few chains that happen to be close to each other. In contrast, MMR-SELECT obtains the highest chain spread and low number of chains, indicating content with low diversity and sparsely presented.

System	Ranking			Chaining		
	Ov↓	I↓	C↓	Spr↓	Den↑	Cov↑
LT5-Casc	1.59	1.56	1.59	**1.93**	1.29	57.12
+MMR-Select	1.50	1.48	1.47	2.36	1.28	53.21
+KvD-Select	**1.41**	**1.46**	**1.44**	2.05	**1.40**	**68.78**
Gold	-	-	-	1.91	1.36	69.65

Table 3: Ranking (left) w.r.t. (Ov)erall quality, (I)nformativeness, and (C)ohesion; and properties of annotated chains (right): spread (Spr), density (Den), and sentence coverage (Cov,%). Best systems are **bolded**. (↑,↓): higher, lower is better.

7 Conclusions

We presented an extractive summarization algorithm that controls each summary quality independently, in scenarios where the input is highly redundant. Redundancy is controlled as the input is consumed, and informativeness and cohesion are balanced during sentence selection.

Results show that our input processing strategy is effective at retrieving non-redundant yet relevant passages, reducing the redundancy levels the rest of the pipeline is exposed to. In addition, our sentence selector emulates human memory to keep track of cohesive chains while building the summary, enforcing ties between noun phrases directly. Extensive automatic and human experiments revealed that it is possible to extract highly cohesive summaries that are as informative as summaries optimizing only for informativeness.

8 Limitations

The proposed system presents the following limitations. First, the system extracts complete sentences and concatenates them to form the final summary. We do not perform any kind of post-editing of discourse markers that might break coherence in the summary. However, our results show that the extracted summaries are still perceived as cohesive by humans. Nevertheless, post-editing is an interesting focus for future work.

Second, we argue about the usefulness of an extractive system in a generative landscape where large language models are predominant. Recent large language models have shown impressive capabilities at producing coherent, assertive text, some even capable of consuming long sequences of tokens. However, hallucinations are a pervasive problem in these systems, especially in highly technical domains like the ones considered in this work. In this scenario, an extractive summary has the advantage of presenting information from the source *verbatim* and hence, without any hallucination. Moreover, extracted summaries preserve the writing style of the input as well as technical, domain-specific terms, avoiding altogether the problems of over-simplification and misstyling.

References

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Intelligent Scalable Text Summarization*.

Regina Barzilay and Noemie Elhadad. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084.

Yimai Fang. 2019. *Proposition-based summarization with a coherence-driven incremental model*. Ph.D. thesis, University of Cambridge.

Murray Glanzer. 1972. Storage mechanisms in recall. In *Psychology of learning and motivation*, volume 5, pages 129–193. Elsevier.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. SNaC: Coherence error detection for narrative summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 444–463, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103.

Mandy Guo, Joshua Ainslie, David C Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736.

Halliday&Rukaya Hassan and M Halliday. 1976. Cohesion in english. *P20*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436.

Sungho Jeon and Michael Strube. 2020. Centering-based neural coherence modeling with hierarchical discourse segments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7458–7472, Online. Association for Computational Linguistics.

Barbara Johnstone. 1994. *Repetition in discourse : interdisciplinary perspectives*. Advances in discourse processes. Ablex Publishing Corporation.

Eileen Kintsch. 1990. Macroprocesses and microprocesses in the development of summarization skill. *Cognition and instruction*, 7(3):161–195.

Walter Kintsch and Teun A van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. Felix: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

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

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213.

Julius Steen and Katja Markert. 2022. How to find strong summary coherence measures? a toolbox and a comparative study for summary coherence measure evaluation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6035–6049, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Maite Taboada. 2004. *Building Coherence and Cohesion: Task-oriented dialogue in English and Spanish*. John Benjamins.

Ana María Vigara Tauste. 1995. Comodidad y recurrencia en la organización del discurso coloquial. In *El español coloquial: actas del I Simposio sobre análisis del discurso oral: Almería, 23-25 de noviembre de 1994*, pages 173–208. Servicio de Publicaciones.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Marilyn A Walker. 1993. *Informational redundancy and resource bounds in dialogue*. Ph.D. thesis, Graduate School of Arts and Sciences, University of Pennsylvania.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3752–3761.

Billy TM Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1060–1068.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18*, pages 5602–5609. AAAI Press.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528.

David Zajic, Bonnie J Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. Discoscore: Evaluating text generation with bert and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3847–3865.

A Dataset Preprocessing and Statistics

For all datasets, we homogenize the source-target length distributions by discarding samples with references that were too short (less than 3 sentences, not usefull for our cohesion analysis) or too long (more than 500 tokens in all datasets except GovReport, for which this threshold is set to 1000). Similarly, samples with short input documents (less than 3 sentences or less than 30 tokens in total) were also discarded. Sentences were re-split using spaCy[4] and trimmed to 100 tokens, whilst sentences with less than 5 tokens were discarded. Table 4 presents the statistics of all dataset in terms of number of tokens.

It is worth noting that we found a discrepancy in PubMed.Text from the '`article`' field (in theory the concatenated sections) would not always have the same text as the '`sections`' field. Hence, we chose data from the '`sections`' field as input document.

Dataset	Input Length Avg.	Input Length Max.	Input Length Q90	Target Len. Avg.
PubMed	3150	119875	5844	206
BigPatent.C	4534	72835	8655	119
GovReport	8840	206622	15752	580
MultiNews	2057	525348	3846	260

Table 4: Dataset statistics in terms of number of tokens showing average, maximum, and 90% quantile (Q90).

B Optimization and Resource Details

Long-T5 models were trained using one NVIDA A100 (80Gb of GPU memory). Table 5 provides a comprehensive account of hyperparameter values used for training and inference in our experiments, for all datasets. The local context extractor is fine-tuned from pretrained Huggingface's checkpoint `google/long-t5-tglobal-base`, whereas the global context encoder is trained from scratch. Finally, Table 5 details the hyperparameter values common to all selectors, as well as selector-specific parameters, optimized w.r.t. each dataset's validation set.

C Complementary Results

In this appendix, we present additional results in terms of metrics and datasets for analysis in §6.

[4] https://spacy.io/

Parameter	Value
Block Selection	
Block length in tokens	2048
Overlapping context size in tokens	200
Damping factor (d)	0.85
Trade-off param. (λ_b)	0.2
Local Context Extractor	
Optimizer	Adam
Learning rate	1E-06
Learning rate scheduler	Const.
Batch size	64
Max. gradient norm	2
Training steps	100 000
Max. input length in tokens	2048
Max. # of sentences extracted	10
Global Contetext Encoder	
# Attention heads	8
# Layers	1
Output layer size	200
Dropout	0.1
Optimizer	Adam
Learning rate	1E-06
Learning rate scheduler	Const.
Max. input length in tokens	16 384
Max. input length in sentences	1000
Batch size	64
Max. gradient norm	1
Training steps	50 000
Sentence Selector	
All selectors. Trade-off param. (λ_{sel})	0.8
Summary budget in number of tokens	
PubMed	200
BigPatent.C	100
GovReport	650
MultiNews	250
KL-Dist. # of histogram bins	40
KvD-Selector.	
Working memory (`WM`)	6
Min. NP cos. similarity (ν)	0.6
Recall cost (γ_{rec})	0.01

Table 5: Hyper-parameter values for all modules in our summarization pipeline.

C.1 Additional Metrics

Semantic Relevance. Table 6 shows BERTScore F_1 scores (Zhang et al., 2019) with importance weighting (IDF) and RoBERTa as underlying model (Liu et al., 2019).

Redundancy. Table 10 presents IUniq and RdRL scores for all systems and datasets analyzed.

Cohesion. The following additional cohesion metrics were explored in preliminary experiments: Extended Entity Grid model (Barzilay and Lapata, 2008), DS-Focus and DS-Sent (Zhao et al., 2023), and RC and LC (Wong and Kit, 2012). However, the values obtained did not show enough expressivity for system-level comparisons and hence, they were not included in the final analysis.

System	PubMed	BigPat.C	GovRep.	MultiN.
Ext-Oracle	88.45	85.84	88.29	88.69
LT5-Flat	**85.71**	**83.77**	86.45	86.03
LED-Flat	83.67	83.04	85.96	85.51
MemSum-Casc	83.52	82.60	85.06	85.07
LLaMa-Casc	82.86	83.17	84.80	85.33
LT5-Casc	85.06	83.66	86.46	85.98
+MMR-Select	85.05	83.66	**86.48**	85.94
+NPassing	85.13	83.67	86.47	**86.01**
+KL-Dist[NP]	85.02	83.52	86.30	85.94
+CCL-Select	84.99	83.63	86.47	85.91
+KvD-Select	84.76	83.35	85.99	85.72

Table 6: Semantic relevance of system summaries in terms of BERTScore F_1.

C.2 Flat Processors and Local Encoders

In addition to LT5-FLAT, we compared against Longformer (Beltagy et al., 2020), trained from the pre-trained encoder module in LED.

Then, we assess the impact of architectural choice for the Local Encoder module in our pipeline by comparing MemSum (Gu et al., 2022), and LLaMA with 7B parameters (Touvron et al., 2023).

The results on informativeness, redundancy, and cohesiveness are presented in Tables 7, 8, and 9, respectively. The following insights can be drawn fro these results. Using LLaMA as local encoder allows our system to select –greedily– sentences that have little lexical overlap between them, prompting low summary redundancy scores and in turn lowering cohesion scores. Moreover, the coverage is severely impacted as seen by the low ROUGE scores. Using MemSum had a similar outcome, although not as severe.

These results might indicate that finetuning a large pretrained model like LLaMA does not necessarily translate to better informativeness, performing much lower than a smaller model pretrained on the summarization task. Perhaps unsurprisingly, task-specific, smaller models can be competitive to massive foundation models trained on 1000x more data.

C.3 Reducing redundancy in block selection

Figure 3 presents the effect of block selection strategies for PUBMED, BIGPATENT.C, and GOVREPORT.

C.4 Effect of Trade-off Parameter λ_{sel}

Figure 4 showcases how summary properties (informativeness, redundancy, and cohesion) vary across

System	PubMed			BigPatent.C			GovReport			MultiNews		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
LED-Flat	40.20	13.87	36.85	36.65	11.07	31.94	57.59	23.40	54.56	45.28	15.78	41.35
LT5-Flat	**48.15**†	**21.45**†	**44.49**†	39.54	**13.25**	**34.30**	59.33	25.94	56.29	**47.07**	**17.54**†	**42.96**
MemSum-Casc	40.29	14.85	37.09	36.07	10.79	30.97	54.91	19.66	51.75	44.47	15.28	40.32
LLaMA-Casc	37.60	11.86	34.51	36.82	11.24	32.00	54.20	19.02	50.90	45.02	15.48	41.00
LT5-Casc	46.16	19.74	42.49	**39.57**	**13.25**	34.26	**59.73**	**26.21**	**56.50**	46.80	17.21	42.66

Table 7: Informativeness in terms of ROUGE F_1 scores (R1, R2, RL), for complementary Flat and Cascaded systems. Best systems are **bolded**. †: System score is statistically different from closest baseline.

System	PubMed		BigPatent.C		GovReport		MultiNews	
	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq
Gold	11.88	19.31	18.11	20.85	13.37	28.78	9.72	16.35
Ext-Oracle	13.91	20.36	14.70	19.51	14.20	29.14	10.08	16.98
LED-Flat	14.70	21.86	17.62	20.07	14.94	31.01	11.25	**19.06**
LT5-Flat	16.49	23.43	19.76	21.32	15.78	32.46	12.24	20.63
MemSum-Casc	12.58	**19.39**	19.41	21.23	13.77	27.47	12.29	19.28
LLaMA-Casc	**11.61**†	19.40	**17.51**†	**18.96**†	**12.43**†	**26.64**†	**10.87**†	19.46
LT5-Casc	17.08	22.94	20.15	21.46	16.34	31.68	12.26	20.59

Table 8: Summary redundancy in terms of sentence-wise ROUGE (RdRL) and inverse uniqueness (IUniq), for complementary Flat and Cascaded systems. For all metrics, lower is better. Best systems are **bolded**. †: System score is statistically different from closest baseline.

increasing levels of λ_{sel}, for all datasets analyzed.

D Human Evaluation Campaigns

In this section, we provide further details about the two evaluation campaigns run. Both campaigns were run on Amazon Mechanical Turk, where Turkers were required to have a Human Intelligence Task (HIT) approval rate higher than 99%, a minimum of 10 000 approved HITs, be proficient in the English language, and have worked in the healthcare or medical sector before. Annotators were awarded \$1 per HIT, translating to more than \$15 per hour. These rates were calculated by measuring the average annotation time per HIT in a pilot study. Furthermore, we implemented the following catch controls: (i) we asked participants to check checkboxes confirming they had read the instructions and examples provided, and (ii) we discard HITs that were annotated in less than 5 minutes.[5] Annotations that failed the controls were discarded in order to maximize the quality. Figure 5 depicts the instructions given to annotators for each campaign.

D.1 Ranking Campaign

We collected three annotations per system-pair comparison and made sure that the same annotator was not exposed to the same document twice. As an additional catch trial, we included in each annotation batch an extra instance with summaries extracted by the extractive oracle and the random baseline.

After discarding annotations that failed the controls, we are left with 708 out of 810 instances (30 documents, 3 system pairs, 3 dimensions, and 3 annotations per pair).

D.2 Chaining Campaign

Participants were shown a single system summary as a list of sentences where tokens that belonged to the same noun phrase were colored the same.Then, the task consists of selecting sets of colored text chunks that belong to the same semantic field. Similarly to the previous study, we collected three annotations per system summary and included the gold summary of an extra system in the campaign.

We collected 908 human annotations of noun-phrase chains for 360 summaries (30 documents, 4 system including gold summaries, and 3 annotations per summary). On average, annotators identified 2.56 groups per summary and 3.49 NPs per group.

E Example Output

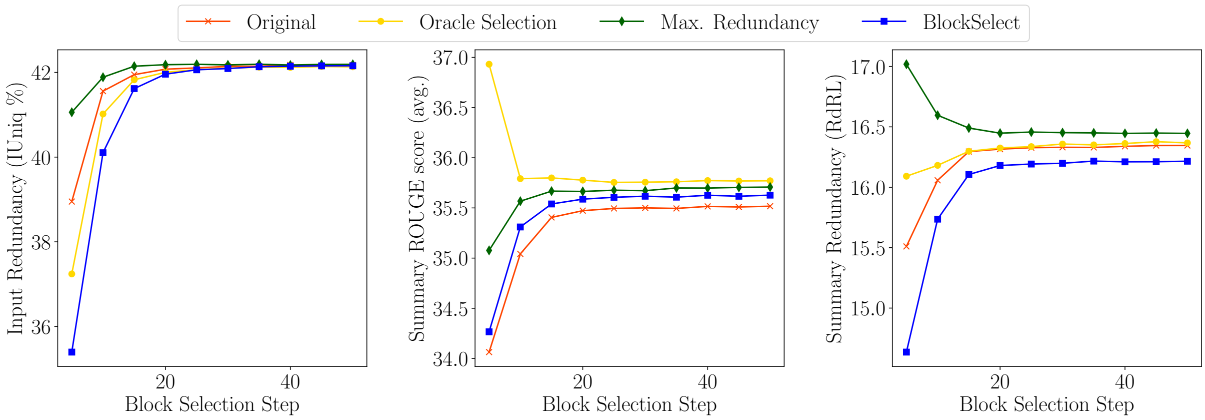
[5]Time threshold obtained from pilot study measurements.

Systems	PubMed			BigPatent.C			GovReport			MultiNews		
	CoRL	**EGr**	**CCL**	**CoRL**	**EGr**	**CCL**	**CoRL**	**EGr**	**CCL**	**CoRL**	**EGr**	**CCL**
Gold	14.45	0.95	0.78	19.19	0.78	0.83	16.21	1.95	0.75	10.45	0.71	0.80
Ext-Oracle	14.68	0.99	0.42	15.94	0.68	0.41	16.55	1.92	0.51	10.85	0.68	0.50
LED-Flat	15.18	1.00	**0.36**	17.67	0.69	0.35	15.06	1.91	0.30	11.31	0.81	**0.30**
LT5-Flat	16.60	**1.10**	0.26	19.76	**0.75**	0.37	16.06	2.00	0.28	12.25	**0.91**	0.26
MemSum-Casc	12.87	0.75	0.25	**20.56**	0.62	0.34	13.93	1.72	**0.29**	**13.16**	0.84	0.26
LLaMA-Casc	12.18	0.70	0.27	17.54	0.70	**0.39**	12.53	1.58	0.28	11.15	0.77	0.25
LT5-Casc	**17.47**	1.07	0.26	20.26	0.73	**0.39**	**16.46**	**2.04**	0.27	12.51	0.90	0.26

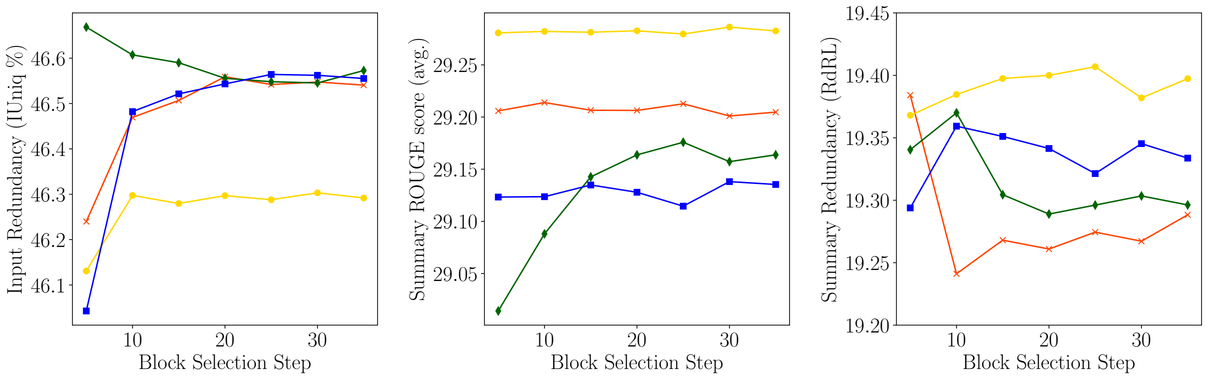
Table 9: Cohesion of extracted summaries in terms of consecutive ROUGE-L score (CoRL) and EntityGraph (E.Gr.), as well as coherence (CCL), for complementary Flat and Cascaded systems. For all metrics, higher is better. Best systems are **bolded**.

System	PubMed		BigPatent.C		GovReport		MultiNews	
	RdRL	**IUniq**	**RdRL**	**IUniq**	**RdRL**	**IUniq**	**RdRL**	**IUniq**
Gold	11.88	19.31	18.11	20.85	13.37	28.78	9.72	16.35
Ext-Oracle	13.91	20.36	14.70	19.51	14.20	29.14	10.08	16.98
LT5-Flat	16.49	23.43	19.76	21.32	**15.78**†	32.46	12.24	20.63
LT5-Casc	17.08	22.94	20.15	21.46	16.34	31.68	12.26	20.59
+MMR-Select	16.99	22.85	19.17†	21.09†	16.16	31.53	12.05†	20.50
+NPassing	16.39†	21.66†	19.79	21.18	16.24	31.42	12.03†	19.92†
+KL-Dist	16.83†	22.08†	20.30	21.44	16.49	31.35	12.57†	20.22
+CCL-Select	16.63†	22.42†	**18.97**†	**20.87**†	16.31	31.65	**11.93**†	20.29†
+KvD-Select	**16.24**†	**21.53**†	21.09†	21.65	16.69†	**30.97**†	12.97†	**19.97**†

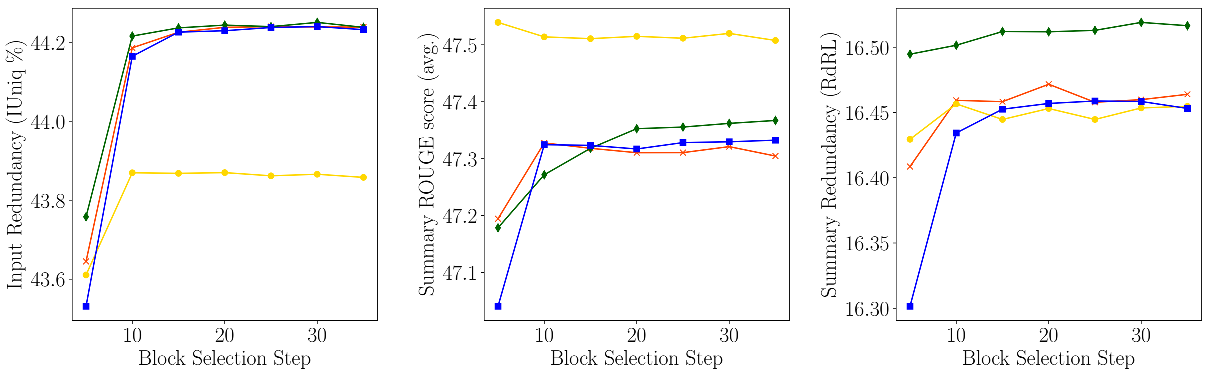
Table 10: Summary redundancy in terms of sentence-wise ROUGE (RdRL) and inverse uniqueness (IUniq). For all metrics, lower is better. See Table 1 for formatting details.



(a) PubMed

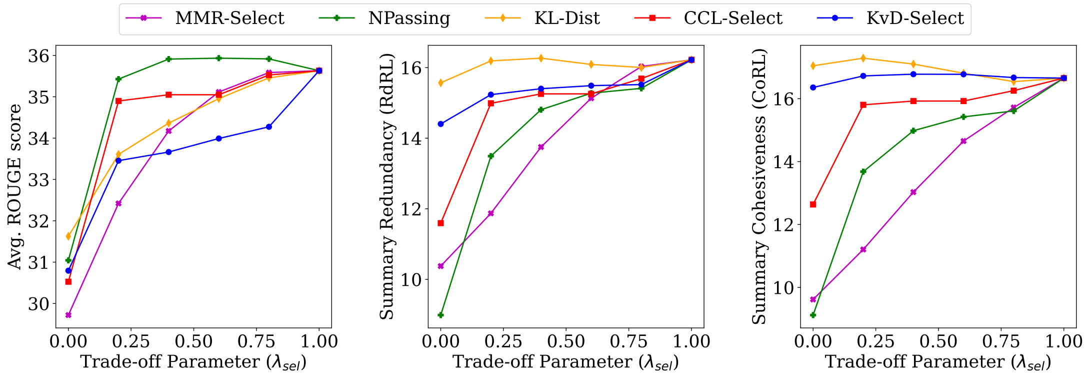


(b) BigPatent.C

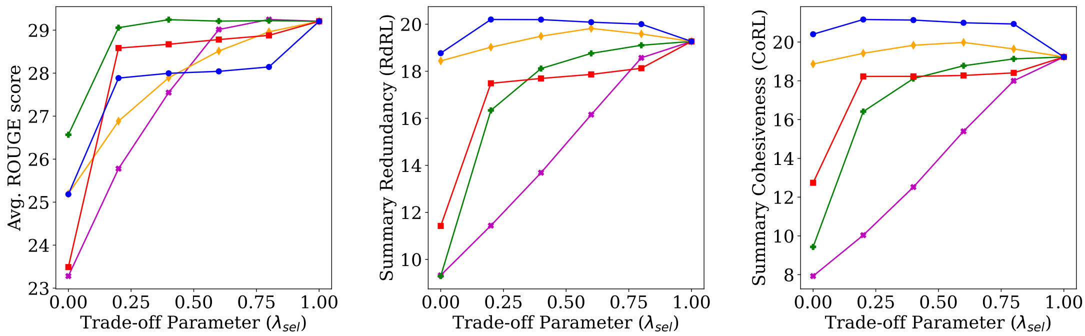


(c) GovReport

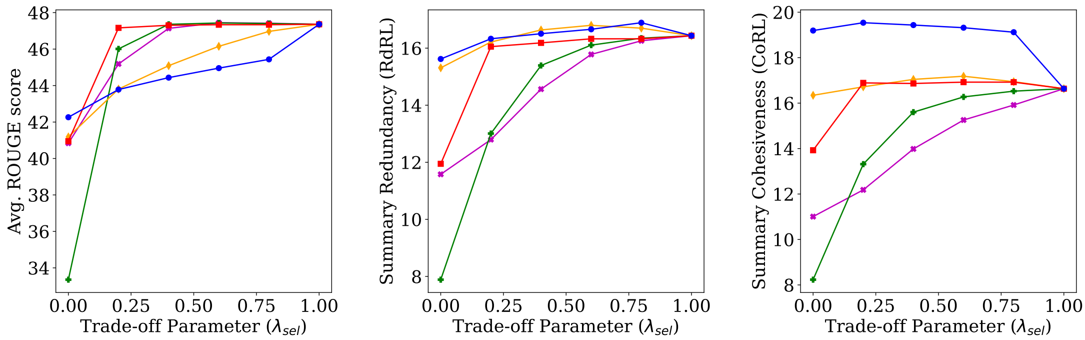
Figure 3: Effect of block selection strategy over input redundancy (left), summary informativeness (mid), and summary redundancy (right), evaluated as block selection proceeds on the validation set of PUBMED, BIGPATENT.C, and GOVREPORT.



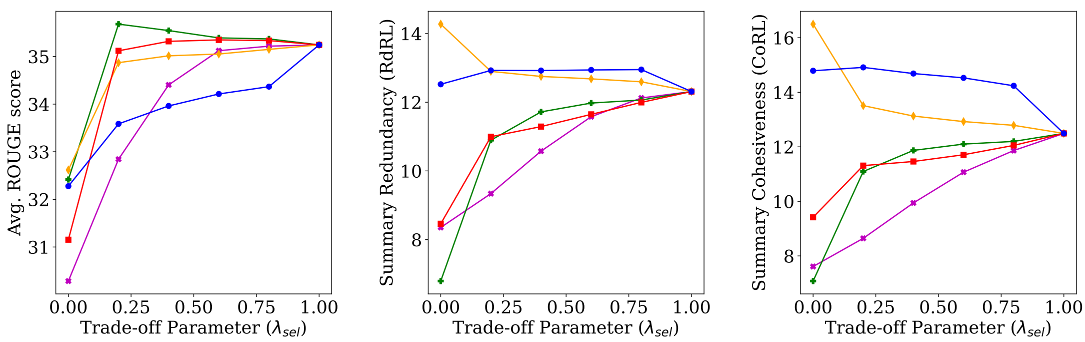
(a) PubMed



(b) BigPatent.C

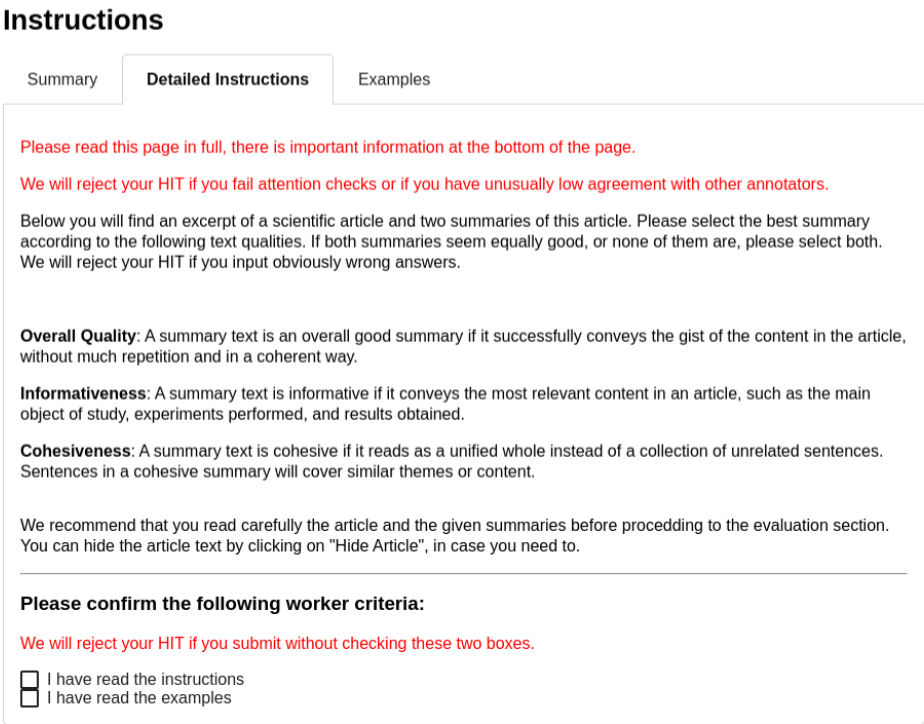


(c) GovReport



(d) MultiNews

Figure 4: Informativeness (left), redundancy (mid), and lexical cohesion (right) across different values of the trade-off parameter λ_{sel} on the validation set of PUBMED, BIGPATENT.C, GOVREPORT, and MULTINEWS.



Instructions

Summary | **Detailed Instructions** | Examples

Please read this page in full, there is important information at the bottom of the page.

We will reject your HIT if you fail attention checks or if you have unusually low agreement with other annotators.

Below you will find an excerpt of a scientific article and two summaries of this article. Please select the best summary according to the following text qualities. If both summaries seem equally good, or none of them are, please select both. We will reject your HIT if you input obviously wrong answers.

Overall Quality: A summary text is an overall good summary if it successfully conveys the gist of the content in the article, without much repetition and in a coherent way.

Informativeness: A summary text is informative if it conveys the most relevant content in an article, such as the main object of study, experiments performed, and results obtained.

Cohesiveness: A summary text is cohesive if it reads as a unified whole instead of a collection of unrelated sentences. Sentences in a cohesive summary will cover similar themes or content.

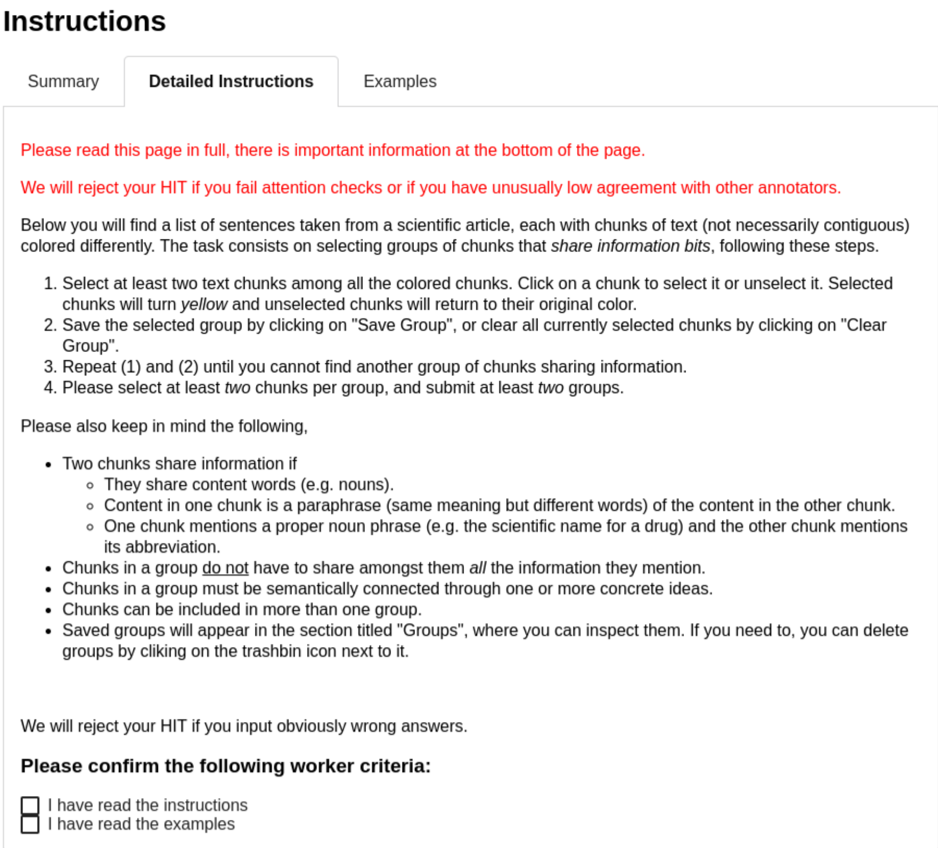
We recommend that you read carefully the article and the given summaries before procedding to the evaluation section. You can hide the article text by clicking on "Hide Article", in case you need to.

Please confirm the following worker criteria:

We will reject your HIT if you submit without checking these two boxes.

- [ ] I have read the instructions
- [ ] I have read the examples

(a) Ranking Campaign



Instructions

Summary | **Detailed Instructions** | Examples

Please read this page in full, there is important information at the bottom of the page.

We will reject your HIT if you fail attention checks or if you have unusually low agreement with other annotators.

Below you will find a list of sentences taken from a scientific article, each with chunks of text (not necessarily contiguous) colored differently. The task consists on selecting groups of chunks that *share information bits*, following these steps.

1. Select at least two text chunks among all the colored chunks. Click on a chunk to select it or unselect it. Selected chunks will turn *yellow* and unselected chunks will return to their original color.
2. Save the selected group by clicking on "Save Group", or clear all currently selected chunks by clicking on "Clear Group".
3. Repeat (1) and (2) until you cannot find another group of chunks sharing information.
4. Please select at least *two* chunks per group, and submit at least *two* groups.

Please also keep in mind the following,

- Two chunks share information if
  - They share content words (e.g. nouns).
  - Content in one chunk is a paraphrase (same meaning but different words) of the content in the other chunk.
  - One chunk mentions a proper noun phrase (e.g. the scientific name for a drug) and the other chunk mentions its abbreviation.
- Chunks in a group do not have to share amongst them *all* the information they mention.
- Chunks in a group must be semantically connected through one or more concrete ideas.
- Chunks can be included in more than one group.
- Saved groups will appear in the section titled "Groups", where you can inspect them. If you need to, you can delete groups by cliking on the trashbin icon next to it.

We will reject your HIT if you input obviously wrong answers.

Please confirm the following worker criteria:

- [ ] I have read the instructions
- [ ] I have read the examples

(b) Chaining Campaign

Figure 5: Instructions given to annotators in the ranking (top) and chaining campaigns (bottom).

System Summary	Chain IDs
Gold (Avg. ROUGE=-; RdRL=8.3; CoRL=6.63)	
Why did Microsoft buy Nokia's phone business?	1,2
We now know Microsoft's answer: the computing giant released a 30-slide presentation today arguing that the move will improve Microsoft's margins on Windows phones, which will allow it to invest more in the platform, which will accelerate sales and market share growth, The Washington Post reports.	1,2,3,5
But John Herrman at BuzzFeed has another explanation: "fear of dying alone."	3
Here's what he and other pundits are saying: the presentation "manages to sound both insane and uninspiring, outlining modest goals that still sound unrealistic," Herrman argues - like capturing a whole 15% of the smartphone market.	2,3,5
"It's a fitting end for the close of Microsoft's Ballmer era, during which the company...missed out on the most important change in consumer electronics in decades" while remaining profitable in unglamorous ways.	1,2,4
Like everyone, Microsoft is trying to ape the Apple model, MobileOpportunity observes.	1,3
But it's not so sure that's a good idea.	3
"There already is an Apple," the blog points out, and other software/hardware hybrid companies, like Palm and BlackBerry, have been crushed under its heel.	1,3
Maybe Microsoft should have tried to patch up its tried-and-true strategy of licensing its OS.	1,2
The move risks complicating Microsoft's crucial relationships with other PC and device manufacturers, one analyst tells ZDNet.	1,2,3
But he adds that "Microsoft needed to make a bold move" or face "certain terminal decline," and that the price it paid for Nokia "seems extremely reasonable."	1,3
Meanwhile, Matthew Yglesias at Slate digs up a fairly interesting memo from Nokia CEO (and, perhaps, Microsoft heir apparent) Stephen Elop, in which he uses the story of a Deepwater Horizon worker leaping from the burning oil platform - a seemingly desperate, yet necessary move - to explain the company's shift from its own failed OS to Windows Phone.	1,2,3,4,11
Of course, Yglesias notes, that move "was basically a total failure."	3,11
MMR-Select (Avg. ROUGE=28.25; RdRL=8.31; CoRL=11.98)	
Summary: Microsoft's acquisition of Nokia is aimed at building a devices and services strategy, but the joint company won't take the same form as Apple.	1,2,10
This crawl was run at level 1 (URLs, including their embeds, plus the URLs of all outbound links, including their embeds).	6
Today's sale price, which includes 1.65 billion euros in patents, is just 5.44 billion euros.	2,7
It's been a rough decade.	-
Microsoft is buying Nokia's cell phone business and licensing its patent portfolio, according to both companies.	1,2
In 2003, Nokia's cell phone market share exceeded 35%.	1,2
That same year, its phone business alone posted an operating profit of 5.48 billion euros.	2,7
Nokia lashed itself to Microsoft's mast after losing out to iOS and Android in the smartphone market share stakes and with the limited success of the Lumia range so far, enough to keep interest in Windows Phone alive, most analysts are seeing a certain amount of inevitability to the acquisition, even if they are split on what its biggest implications are.	1,2,8,10
The seed for this crawl was a list of every host in the Wayback Machine.	6
The WARC files associated with this crawl are not currently available to the general public.	6
Five years ago was the year the App Store first opened.	2
Windows Phone has barely dented the now much larger smartphone market.	2
Many at the time wondered if Stephen Elop's time at Nokia would be spent grooming the company for purchase —a foreigner in all possible ways, he began his time at the company with a memo rightly but offensively declaring Nokia's proud platform a failure and quickly pledged the company's commitment to the still-tiny Windows Phone.	1,2,4
KvD-Select (Avg. ROUGE=26.33; RdRL=12.48; CoRL=14.41)	
Summary: Microsoft's acquisition of Nokia is aimed at building a devices and services strategy, but the joint company won't take the same form as Apple.	1,2,10
Microsoft has been working on its evolution into a devices and services company, moving away from the services business it has traditionally been, for several years now with limited success.	1,2,8
Nokia lashed itself to Microsoft's mast after losing out to iOS and Android in the smartphone market share stakes and with the limited success of the Lumia range so far, enough to keep interest in Windows Phone alive, most analysts are seeing a certain amount of inevitability to the acquisition, even if they are split on what its biggest implications are.	1,2,8,10
Owning the desktop (via Windows) and building additional services on top, like Office or Search, has been vital for Microsoft's strategy until now, so, as our interest shifts from the desktop to the tablet or smartphone, it's essential to Microsoft's broader business (even Azure) that it can retain that connection in some form.	1,2,9
But he said Microsoft's challenge remains how to unite the myriad services and brands - Windows, Nokia, Live, Surface, Xbox, Bing, and more - into a cohesive experience that will command and cement customer loyalty.	1,2,9
It felt like a radical about-face, but no matter: Nokia and Microsoft were going to save each other.	1,9

Table 11: Reference summary, along with summaries extracted by MMR-SELECT and KVD-SELECT for a MULTINEWS sample with informativeness (average ROUGE score), redundancy (RdRL), and cohesion (CoRL) scores. Each sentence is annotated with lexical chains, color-coded in the text and IDs shown to the right. Text was detokenized and truecased for ease of reading.