Multimodal Summarization of Complex Sentences

Naushad UzZaman
Computer Science Department
University of Rochester
naushad@cs.rochester.edu

Jeffrey P. Bigham
Computer Science Department
University of Rochester
jbigham@cs.rochester.edu

James F. Allen
Computer Science Department
University of Rochester
james@cs.rochester.edu

ABSTRACT
In this paper, we introduce the idea of automatically illustrating *complex sentences* as multimodal summaries that combine pictures, structure and simplified compressed text. By including text and structure in addition to pictures, multimodal summaries provide additional clues of what happened, who did it, to whom and how, to people who may have difficulty reading or who are looking to skim quickly. We present ROC-MMS, a system for automatically creating multimodal summaries (MMS) of complex sentences by generating pictures, textual summaries and structure. We show that pictures alone are insufficient to help people understand most sentences, especially for readers who are unfamiliar with the domain. An evaluation of ROC-MMS in the Wikipedia domain illustrates both the promise and challenge of automatically creating multimodal summaries.

Author Keywords
Multimodal summarization, summarization, visualization, illustration, picture, text-to-picture, automatic illustration, sentence compression, pictorial representation, AAC, augmentative and alternative communication, ROC MMS.

General Terms
Algorithms, Experimentation.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous; I.2.7 [Artificial Intelligence]: Natural Language.

INTRODUCTION
Pictures, diagrams and illustrations are included in manually-created text because they help people comprehend and remember information [1]. Including alternative, supportive representations of text might help people with reading difficulties understand text better, for instance those reading text not in their first language, children, older adults, or people with cognitive disabilities. Unfortunately, creating illustrations is expensive and time-consuming, and consequently most text has only a few illustrations, if any at all. In this paper we introduce ROC-MMS, a system that automatically converts existing text to multimodal summaries (MMS) that capture the meaning of a complex sentence in a diagram containing pictures and simplified text related by structure extracted from the original sentence.

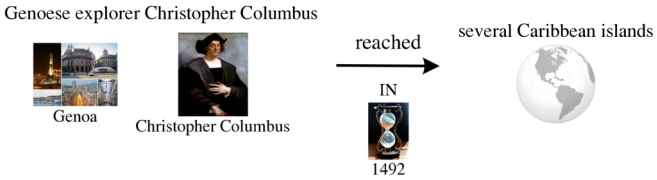

Figure 1: Multimodal summary (MMS) of the sentence, *"In 1492, Genoese explorer Christopher Columbus, under contract to the Spanish crown, reached several Caribbean islands, making first contact with the indigenous people."*

Motivated by sayings like, "A picture is worth a thousand words" prior work on Automatic Illustration and Text-to-Picture synthesis has approached the very difficult problem of generating pictorial replacements for text. Although this is an interesting challenge, existing systems have generally found success only within the domain of simple sentences of the type found in children's books [2-4]. The problem of multimodal summarization relaxes the problem by allowing text to augment pictorial and structural information.

Automatic Illustration is inherently difficult. To understand the problem better, we initially asked two annotators[1] to identify the main idea[2] (main event) and related entities (subject, object, etc) from sentences and find representative pictures. Sentences were chosen from the Wikipedia entries *United States* and *France*, and annotators were asked to include Wikipedia pictures in their illustrations. The annotators reported that it was too difficult to illustrate 19.59% of the entities using Wikipedia pictures and thought

IUI'11, February 13–16, 2011, Palo Alto, California, USA.

[1] Annotators are graduate students and not among the authors. Their annotations were used as a gold standard in our evaluation.

[2] In this paper, we loosely interchange between main idea, main concept and main event.

that 15.08% of entities couldn't be represented with pictures at all (e.g. "territory", "height of power", "French War of religion", etc and temporal expressions in general). These results suggest that it will often be difficult to find appropriate pictures and some entities are inherently unable to be illustrated easily with pictures. It can be particularly difficult to represent entities in an unfamiliar domain. For instance, if someone doesn't know how Christopher Columbus looks like, even a good picture of Christopher Columbus will only convey general attributes (man, possibly historical).

To remedy this problem MMSs keep both images and representative text, unlike previous systems for automatic illustration [2-6]. In this way, we can handle cases lacking a good picture and address cases that are hard to illustrate. Presenting pictures and text together can also improve both the understanding and remembering of concepts. According to dual code theory [7], text and pictures result in two different kinds of conceptual representations. These representations may allow independent access to information and hence benefit retention. Picture and text repeat important information, and may have similar beneficial effects on memory as explicit repetitions [8, 9]. Processing the information twice, once as text and once as a picture, may facilitate comprehension and memory. Finally, pictures often have a motivating effect, and text with pictures may also be more enjoyable to read, since the reader does not have to work as hard to understand the text and pictures also facilitate better comprehension of the text broadly beyond what is illustrated [10]. So our decision for inclusion of text with pictures is backed by theories that support that it helps people for better understanding and memorizing.

To keep the MMS representations simple and easy to process, we simplify text so that it retains only the most important information, instead of the full text. We define the most important information as the subject (who did it), the event (what action), object (to whom or what) and prepositions directly related to the subject, main event, or object (how). This effectively converts complex sentences into simpler sentences. In this way, the reader can read out the text as a simple sentence in addition to seeing the pictorial view, making it easier to remember and understand text, and relate it to the full, complex text if they choose, such as when searching for details abstracted out of the MMS view.

MMS can potentially help a diversity of readers. For example, highly-capable readers may use MMS to skim content or understand content more easily. The alternative, simplified representation it provides may be useful for children who are learning to read and for second language learners, as seeing pictures together with text may enhance learning [11]. Furthermore, it has been previously shown that when one component of the reading process is dysfunctional, other compensating skills may become highly developed [12]. It is estimated that more than 2 million people in United States have significant communication impairments that led them to rely on methods other than natural speech alone for communication [13]. Automatic Illustration of texts may eventually help these people understand text better. Automatic illustration can also help to support other representations like Pictorial Temporal representation [14] or can be paired-up with screen reading applications [15], which could further benefit people who have problems reading by allowing them to see content in multiple forms while listening to it being read.

We define *multimodal summarization* of complex sentences as the combination of illustrations and a compressed form of the sentence text in simple sentence structure. In the next section we will describe the challenges for multimodal summarization and describe related work for the required subtasks. We then describe ROC-MMS, our system for multimodal summarization and describe an evaluation of it. Finally, we discuss potential for future work.

SUBTASKS AND RELATED WORK

Multimodal summarization (MMS) of complex sentences gives readers the main idea of the sentence using pictures and compressed text structured as simple sentence. Creating MMSs is challenging and involves many subtasks. In this section, we will describe each of the subtasks and the related work for each subtask, and the approach taken in ROC-MMS. The general steps in the MMS approach are the following:

1. Identify both the main idea of the sentence and related entities and use them to create a compressed summary
2. Extract pictures for the entities.
3. Add structure to the pictures and text.

Identifying the main idea and related entities

Natural language sentences often convey multiple ideas, but representing multiple ideas with pictures can quickly become confusing. We, therefore, chose to express only the main idea of a sentence with MMS. If readers can understand the main idea of the sentence, then they may be able to later use the original text to decipher further details.

The subtask of identifying the main idea of the sentence itself has two components. First, the important idea (the main event or main action) must be extracted, and, second, the entities related to the main idea need to be extracted, as illustrated in the following example drawn from Wikipedia:

"In 1492, Genoese explorer Christopher Columbus, under contract to the Spanish crown, ***reached*** *several Caribbean islands, making first contact with the indigenous people."*

The summary or compressed form of the sentence is "Christopher Columbus ***reached*** several Caribbean islands in 1492." Hence, the main event or main idea in the sentence is ***reached*** and the entities related to the event

reaching are Christopher Columbus (subject), several Caribbean islands (object) and 1492 (preposition in).

A similar problem already addressed in the natural language processing community is called sentence compression [16]. In sentence compression, unnecessary information is removed while retaining the grammaticality of the sentence. Sentence compression might remove related entities of main event in the process of removing unnecessary information. This approach also doesn't give a simple sentence structure.

Another approach is main event extraction using the TimeML annotation scheme [17]. In this scheme, the *main event* label corresponds to the main idea of the sentence. Most competitive systems use syntactic and semantic information and machine-learning classifiers to identify events. For an overview of recent systems in this area, see the results of TempEval-2 [18]. The main events are annotated as part of the TempEval-2 task, although results on identifying main events were not explicitly reported.

In the literature on Automatic Illustration for extracting entities, a popular approach has been to first extract representative keywords and then generate images for these keywords [6]. Keyword extraction has been studied in the natural language processing/information retrieval community [19, 20]. Goldberg et al. [2, 4] extract actions (events), who did them and to whom. They don't focus on identifying only the important idea (action) because their experimental domain only contains short and simple sentences (and are, therefore, unlikely to contain more than one event). They convert the problem of identifying entities to a sequence labeling problem and use Conditional Random Fields for classification. On the other hand, Mihalcea and Leong [3] do not try to extract the entities, but they extract the pictures word-by-word and represent them linearly. Both approaches work best on simple sentences in which order roughly matches the role of the extracted entities. The ROC-MMS system includes a full natural language parse of the complex sentence in order to extract entities regardless of the order in which they appear.

Extracting Pictures for Text

Once we have the event and related entities, we next extract pictures to represent each concept. The task of associating words to pictures is similar to image retrieval. Although some work uses computer vision techniques for retrieval, most work (including popular image search engines) rely primarily on the text found near images in documents to find general images [21]. ROC-MMS generally follows this approach as well, but uses additional information automatically generated from the structure of the sentence to weight its search terms.

Text-to-scene conversion places objects in 3D environment and is intended to aid graphic designers. This usually works with detailed descriptive text with visual and spatial elements. One of the best-known systems of this kind is WordsEye [22]. They are usually not intended as assistive tools to communicate general text, because in that domain the texts are usually explaining the situation like *"the house is 7 foot tall with two glass window and a door"* and the system will try to interpret the natural language and create the 3D environment of the described situation. In contrast, we want to take a sentence from an existing news source, Wikipedia, or a book and represent it with pictures to help people to understand the text better.

Barnard and Forsyth [23] introduced the idea of auto-illustration as inverse of auto-annotation. Joshi et al. [6] approached this problem by considering the pair-wise reinforcement based on both visual and WordNet-based lexical similarity. This work identifies a few representative pictures for a story, which has practical applications like identifying representative pictures for news articles, or different articles, but not appropriate for our problem.

Goldberg et al. [2, 4] built their own database of images to use for certain text and if they couldn't find any appropriate image in their database then they do web image search and apply some vision techniques to identify the appropriate picture. Mihalcea and Leong [3] use an in-house image database, PicNet and other resources[3].

Adding Structure to Improve Understanding

Having identified pictures and compressed text, the final step is to combine these elements in a layout structurally representative of what happened, who did it, to whom and how. To our knowledge, the only other work that attempts to address this problem is Goldberg et al. [2]. Their system identifies "who", "what action" and "to whom" by converting the problem into sequence labeling. They propose a layout represented by the sequence ABC, where A represents who did the action, B is what action was done and C is to whom. An example output of their system for *"The girl rides the bus to school in the morning"* is below:

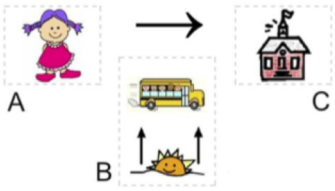

Figure 2: Example output of [2] illustrating the labeling of sequences where each element is assigned a picture.

In this work, the textual information is ignored and represented only with pictures. Images incorrectly extracted in the previous step may confuse people more than helping them because there is no additional information to guide them to the correct interpretation. MMS includes extracted text in case of errors. With both picture and compressed text, we can represent hard-to-depict, but important, entities with text that may be ignored by prior work. We do not attempt to represent events (the action) with a picture, since this is a much more challenging task.

[3] http://tell.fll.purdue.edu/JapanProj/FLClipart/

This work also tries to identify the A (who), B (what action) and C (to whom) of their ABC layout by converting it to a sequence-tagging problem, which is well studied in NLP [24]. The problem with that approach is the requirement for hand-labeled training data, which will be a barrier for adaptation of the solution to a different or more complex domain. ROC-MMS uses dependency parsing to identify similar dependencies or related entities, without needing the hand-annotated training data.

Finally, they restrict their attention to single simple sentences and their experiments were on domains that use very simple English, such as short narratives written by and for individuals with communicative disorders; one-sentence news synopses written in simple English targeting foreign language learners; and the child writing sections of the LUCY corpus. For complex sentences, they anticipate the use of text simplification to convert complex text into a set of appropriate inputs for their system. It is not clear how well they can eventually represent the complex sentences in their layout, since they are not considering "how" something happened.

ROC-MMS addresses these problems for unrestricted texts that include complex and compound sentences.

ROC-MMS

In this section we will describe ROC-MMS, and how it approaches the subtasks described in the previous section.

Identifying the main event(s)

ROC-MMS finds concepts by identifying the *events* and related entities, and then identifies the main event to identify the main concept or the main idea.

Event extraction

Our view for *event* matches with the TimeML temporal annotation scheme [17], which considers *events* a cover term for situations that *happen* or *occur*.

ROC-MMS extracts events using the TRIOS system [25], which had a very competitive performance in the TempEval 2010 task for temporal information extraction [18]. The TRIOS system first parses text with the TRIPS parser [26] and uses hand-coded rules to extract events. The extraction rules are tuned for high recall and identify many more events than is necessary, including a few non-events. In the next step, a classifier is used as a filter to remove unnecessary events.

The main event identification classifier takes all events for a sentence as input and identifies the main event from the sentence. In one of the tasks for TempEval 2010, main events were labeled. We used that labeled data to train our main event classifier. For this classification task, we used an off-the-shelf Markov Logic Network classifier (*thebeast*)[4]. As features, we used lexical features (word, stem, next word, previous word, previous verbal word sequence), syntactic features (part-of-speech tag, tense, voice, polarity, TimeML aspect, modality, pos sequence, previous verbal pos sequence, next pos, previous pos) and semantic features (abstract semantic class – ontology type, TimeML class, semantic roles and their arguments) of events. The syntactic and semantic features are mostly generated from TRIPS parser output and also using other classifiers.

This classifier first identifies the main events from the sentences. Then we run another pass to make sure every sentence has at least one main event. We force every sentence to have a main event. If a classifier didn't identify a main event in a sentence, then we consider the first *verbal* event as the main event of the sentence. We back off to the first verbal event because it has a high baseline performance for the main-event identification task.

Extract entities related to the event

Instead of extracting all entities in the sentence [3], we extract only those entities related to the main event. We use the relations between the event and the related entities in the next step to structure them. From the parsed representation created from the Stanford dependency parser[5], we find dependencies[6] in order to extract the subject (`nominal subject - nsubj, agent`), object (`direct/indirect object - dobj/iobj, passive nominal subject - nsubjpass`) and other dependencies (`prepositions`). For easier representation, we cluster all prepositional modifiers into a single entity, but include the preposition when representing.

An example will help to illustrate how we use the dependency output to extract related entities for the events. The following is the Stanford dependency parser output for the sentence, *"French fur traders **established** outposts of New France around the Great Lakes."*

amod(traders-3, French-1)
nn(traders-3, fur-2)
nsubj(established-4, traders-3)
dobj(established-4, outposts-5)
nn(France-8, New-7)
prep_of(outposts-5, France-8)
det(Lakes-12, the-10)
nn(Lakes-12, Great-11)
prep_around(established-4, Lakes-12)

The main event here is ***established***, the subject is *traders*, the object is *outposts* and the preposition (around) is *Lakes*. By propagating through `nn (noun compound modifier)` and `amod (adjectival modifier)` dependencies, we extract the following entities: (subject: "*French fur traders*"), (object: "*outposts*") and (preposition: "*Great Lakes*"). For subject, object and prepositions, we propagate through the `nn` and `amod` in this way and extract

[4] http://code.google.com/p/thebeast/

[5] Stanford dependency parser: http://nlp.stanford.edu/software/lex-parser.shtml.

[6] Details on dependencies: http://nlp.stanford.edu/software/dependencies_manual.pdf

the resulting entities. The next step is to find the representative pictures for the entities. If we fail to find an image for any entity, we propagate through all dependencies (instead of just `nn` and `amod`) to extract an *entity phrase*. For example, we would extract the phrase "outpost of New France" for the object and "the Great Lakes" for the preposition, in the above examples. We then search for the picture of the entity phrase, instead of the entity. These steps are described in more detail next.

Extracting Pictures for Concepts

Image retrieval is a complicated task, even for humans because what constitutes a *representative image* is subjective. As a result, we simplified the problem by restricting our image search to Wikipedia, which we have found to often produce appropriate images. This has the following two benefits: (i) pictures of an entity are often found on the wiki page for that entity, and (ii) Wikipedia articles often have info box pictures selected by human editors that are often correct and representative.

Finding pictures for an event ("what action" according to [2]) is much harder. When humans are asked to find pictures for events, they will often search for the event along with subject or object. For example, for the event "conquered" in the context "*Rome conquered the Gauls*", an appropriate image would likely include Roman soldiers (it would be even better if it somehow indicated that the conquering occurred in Gaul). Search results for *conquered* alone include the following images in the top results:

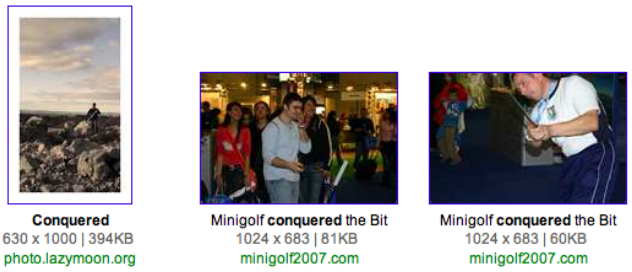

Figure 3: First three results from Yahoo Image Search for the word "conquered" illustrating the difficulty in finding good representative pictures even for simple concepts.

A useful heuristic for finding better representative images is therefore to concatenate the action with the subject and object (if available, or just subject or object, if the other one is not available). Often web image search results still do not return the most appropriate images for our use as the first result. This can be fine for humans, who may glance through the top few results and pick the most appropriate one. Restricting pictures only to Wikipedia is a simple way to produce better results.

Our methods for identifying the pictures are described below with different modules.

Module find_image_in_wikipage(wikiurl):

(i) Find the infobox picture

(ii) If infobox has multiple pictures, then consider the picture with largest width[7]

(iii) If there are no infobox picture

a. Find all images

b. Tokenize the image filename[8] with "_", ",", "[A-Z]", and spaces as delimiters

c. For each image

i. Find the edit-distance between tokenized filename and each word in wiki article name

ii. Sum all scores, that's the relatedness score for an image

d. Return the picture with highest score and the score

Module find_page_and_image(query):

(i) Search with "wikipedia " + query using yahoo search api[9]

(ii) Keep only en.wikipedia pages

(iii) Traverse the resulting wiki pages one by one

(a) Get the representative image with score from the wiki page's url using the module: *find_image_in_wikipage(result page)*

(b) If the resulting image's score is above threshold (we used 1.0) then return the image

Module sentence_to_images(sentence):

(i) Extract events, main event and the entities and entity phrases related to main event (all these described in previous section)

(ii) For each of the dependencies (subject, object, prepositions):

(a) If any word forms a main Wikipedia entry:
Find the image in those wiki urls using *find_image_in_wikipage(wikiurl)*

(b) If no result found so far and the entity doesn't have a wiki link
Then find the image using yahoo search with *find_page_and_image(entity)*

(c) If no result found so far and any word in the *entity phrase* is linked to wiki urls:
Then find the image in those wiki urls using *find_image_in_wikipage(wikiurl)*

(d) If no result found so far and entity phrase doesn't have a wiki link:

[7] We found that when there are multiple pictures then the larger width picture is usually the main representative picture.

[8] We are only considering the tokenized filename, because, i. wikipedia has very descriptive image filenames, ii. text descriptions next to images are not consistent, some pictures have lots of text and others don't have any, since sometimes it's just neglected by contributors, if the wiki entry is not too interesting. But we consider the alt tags of images, which is also very sparse. So we give a lower weight for that score (we used 0.25 for alt tags and 1.0 for image filename score).

[9] http://developer.yahoo.com/search/web/V1/webSearch.html

Then find the image using yahoo search with *find_page_and_image(entity phrase)*

Consider the following clarifying example. The input sentence from Wikipedia is "*French fur traders* ***established*** *outposts of New France around the Great Lakes.*" (Underlined words are links to other Wikipedia pages). ROC-MMS extracts the following main event (in this case, the only event) as *established*, and the extracted entities and entity phrases are: (subject: *French fur traders)*, (subject phrase: *French fur traders*), (object: *outposts)*, (object phrase: *outposts of New France)*, (preposition: around – *Great Lakes)*, (preposition around phrase: *the Great Lakes*). First consider the subject, *French fur traders*. "Fur traders" has a wiki link, but the page does not have an infobox. For images on the linked page, we find the edit distance between the tokenized filename and the article name (Fur trade) and the best image according to the process described previously.

Next we consider the object *outpost*, which does not have a wiki link. We search using Yahoo! restricting to Wikipedia pages, which doesn't return any images above threshold in first 10 resulting pages. We then check the object phrase – *outposts of New France*, and *New France* has a wiki link, and we find a representative picture from that link.

In our algorithm, we search for the entity first, instead of checking wiki URLs in the entity phrase, because sometimes in Wikipedia contributors fail to tag entities to its wiki article. For those cases, our *yahoo_search* module finds the expected wiki article. So we try this step first and if it fails, then we check the wiki links in the entity phrase, as shown in this example. Finally, the preposition (around) is Great Lakes, which links to its wiki article and we get the representative picture for that too.
If there are multiple wiki links in an entity (or entity phrase) then we find images from all wiki links and cluster them.

Figure 4: Clustered image of Genoa and Christopher Columbus for entity *"Genese explorer Christopher Columbus".*

We also cluster all prepositions. The sentence *"The modern name 'France' derives from the name of the feudal domain of the Capetian Kings of France around Paris"* contain two prepositions, *from* and *around.* We extract pictures for *from* the *name of the feudal domain of the Capetian Kings of France* and also for *around Paris*, and then combine them.

Figure 6: Example of clustering prepositions.

Our annotators were unable to find images to represent temporal expressions, and indeed this is a difficult problem. To handle that problem, we give special treatment to temporal expressions. To identify temporal expressions, we use the TRIOS temporal expression identification and normalization system[10] [25], which had the second best performance in TempEval-2 [18]. When we identify a time, instead of searching for a picture of it, we represent it with something that represents time and add the text below. One example is given below.

Figure 5: The representation of a temporal expression includes the extracted text and a picture. The picture conveys time generally, but not a specific time.

Structuring the images and compressed text

The final step is to combine the image and compressed text into a structured format[11]. Every sentence has a main event, which we don't try to represent with pictures, a subject entity, object entity and clustered prepositions. We construct MMS using the following visual layout of these elements.

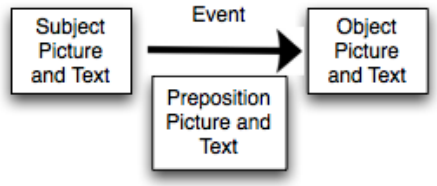

Figure 7: Generalized visual layout for MMS.

This representation is very similar to ABC layout [2], since the subject and object are essentially who did the action and to whom, however the primary difference is that MMS

[10] The temporal expression normalizer is also available as open source at: http://www.cs.rochester.edu/u/naushad/temporal

[11] All our auto-generated diagrams are generated using GraphViz toolkit.

includes prepositions and does not attempt to find a picture for the main event. As mentioned earlier, it is not clear from the description how they represent hard-to-depict events. It might have worked in their simple domain; however, they explained they only find pictures for easy-to-depict words. Many events can be missed as part of the filtering process. ROC-MMS makes appropriate trade-offs that enable it to create MMS diagrams for arbitrary text, even text that includes complex sentences.

One example output from our system is given below:

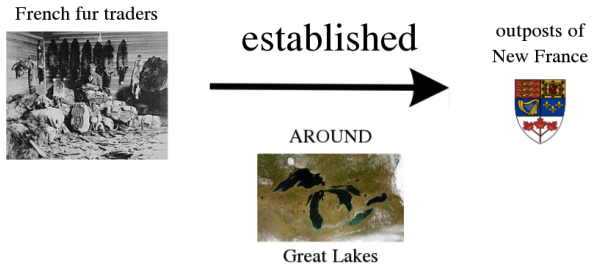

Figure 8: Multimodal summary (MMS) of the sentence, *"French fur traders established outposts of New France around the Great Lakes; France eventually claimed much of the North American interior, down to the Gulf of Mexico."*

Some sentences do not contain prepositions (or the they may not be correctly extracted). In such cases, we show only the event, subject and object, as shown below.

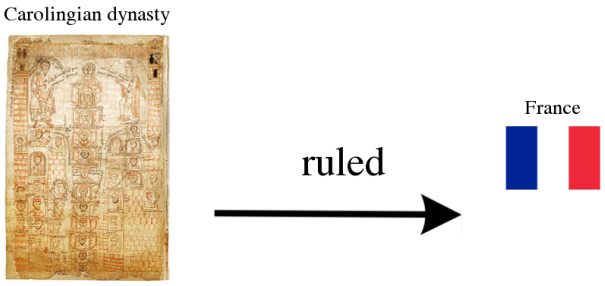

Figure 9: MMS of the sentence, *"The Carolingian dynasty ruled France until 987, when Hugh Capet, Duke of France and Count of Paris, was crowned King of France."*

For sentences lacking an object, we merge the event text with the subject text and show it in subject text field. In the following example, *died* (event) is merged with the *Charles IV* (subject).

Figure 10: MMS of the sentence, *"Charles IV (The Fair) died without an heir in 1328 ."*

EVALUATION

Illustrating a sentence with a diagram of pictures and text is difficult; evaluating how good a diagram is may be even harder because it is very subjective. In this evaluation section, we first evaluate the subtasks of our multimodal summarization system in isolation. We then evaluate how well our representation retains the overall information of the overall sentence. All our evaluations are done on 44 sentences drawn from Wikipedia article on United States and France.

Identifying the Main Event and Related Entities

We trained our main event identification classifier on TempEval-2 training data and tested it with 10 cross validation. Our performance for main event identification was around 77.94% (fscore). The baseline of choosing the first verbal event as the main event achieves around 59.64% on the TempEval domain. We ported that system on the Wikipedia domain and evaluated considering each annotator as gold standard. We calculated precision and recall for both cases, the performance is reported in Table 1.

Metric	Performance
Precision	79.10%
Recall	73.11%
Fscore	75.98%

Table 1. Main event identification performance

We extract entities by first traversing the nn *(noun compound modifier)* and amod *(adjectival modifier)* dependencies of the dependency tree. If that entity results in a good picture (the matching score is above threshold), we keep it; otherwise we traverse through all dependencies of the event, resulting in a phrase. Our extracted entities often don't exact match with the annotator's entity but may partially[12] match with them. We report the average performance (considering both annotators) of our system on entity extraction in Table 2. We only consider cases in which our system and the annotators identified the same main event.

Metric	Performance
Average strict precision	29.29%
Average strict recall	31.64%
Average relaxed precision	76.76%
Average relaxed recall	83.82%

Table 2. Entity extraction performance

Extracting Pictures

For evaluating how well our system extracts pictures, we compared our system output to extractions by two human annotators. We consider cases where our system and the annotater, with relaxed matching, identified the same main event and same entities and both extracted an image. In

[12] Either our entity is substring of annotator's entity, or vice versa. Relaxed matching is partial matching.

Table 3, we show the percentage of cases when both systems extracted an image, given that both systems extracted the same entity. Not all extracted entities have a picture because human annotators sometimes didn't extract the picture because they thought some concepts couldn't be illustrated with a picture and sometimes thought there were no suitable pictures in Wikipedia to represent that entity. We also didn't suggest a picture for entities if no picture was found with a score above threshold. We compared between two annotators and show the average system performance. We can see that our system has a very similar performance compared to performance between each annotators.

Evaluation	**Both entity got Image**
Annotator1 vs Annotator2	66.66%
Average of Annotators vs System	65.47%

Table 3. Performance of Image Extraction

On these selected matching pictures, we compare our extracted image with the images extracted by the annotators. We classify our output into *Same Image* (if both the system and annotators extracted the same image), *Different Image* but acceptable (e.g. for France, one extracted the French flag and the other extracted a map of France) and finally *Bad Image* by our system (this category is the category of images that we think are not acceptable, i.e. wrong representation of the text). A judge, another graduate student - who was not an annotator or an author, performed this classification.

Evaluation	**Ann 1 vs Ann 2**	**Ann vs System (Average)**
Exact same image	47.05%	21.51%
Different image, but acceptable	52.95%	44.15%
Different and bad image		34.34%

Table 4. Performance on quality of our extracted images

We can see that our system extracts decent pictures around 65% of the time.

How well our structure with simple compressed text helps to understand text better

In the previous subsections, we showed our performance in the different subtasks, which eventually propagates to the final performance; but overall how well does our system generate diagrams that convey the message of the content to the users? Does automatic illustration really help text comprehension? Do human-generated illustrations help for text comprehension? An illustration without text is unlikely to be useful if the domain is new to the reader because the reader won't be able to interpret the pictures in the first place. That's why MMS diagrams include simple compressed text and the simple structure along with the event, subject, object, and prepositions.

In this section, we motivate MMS over picture-only diagrams by showing that users get a better understanding from the MMS diagrams generated by ROC-MMS than they do for diagrams containing only pictures, even when human annotators have identified the pictures.

For this evaluation, we recruited participants on Amazon Mechanical Turk[13]. In the task shown to participants, we show our system generated MMS diagram and ask the turkers to explain the diagram in English text. Participants were also given the option of saying that they "Can't explain the diagram." One example is shown in Figure 11.

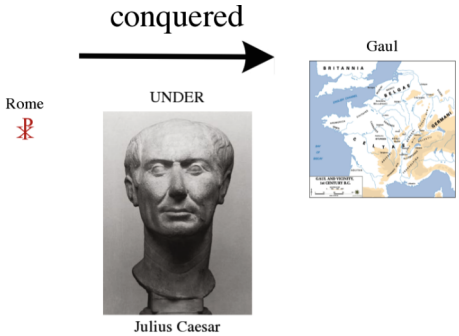

Figure 11: ROC-MMS generated diagram for *"Gaul was conquered by Rome under Julius Caesar in the 1st centiry BC"*

Next we created the diagram using entities and pictures selected by human annotators (representing a gold standard), but we didn't add the structural layout or text like our MMS diagram. Influenced by Mihalcea and Leong [3], our baseline ordered the picture of the entities in the order of the sentence. For example, for the sentence, *"Gaul was **conquered** by Rome under Julius Caesar in the 1st century BC"*, we created the diagram with first picture for Gaul then event conquered (in text), then picture for Rome and finally Julius Caesar. The annotators thought *1st century BC* was hard to illustrate, and so did not find a picture for it. We asked our annotators not to find pictures for events, since we are not going to represent events with pictures and added the text for *events* instead in annotator's diagram. One example diagram is shown in Figure 12.

Figure 12: Diagram using human identified entities and pictures for *"Gaul was conquered by Rome under Julius Caesar in the 1st century BC"*

Although the pictures are accurate, it is quite difficult to find the meaning of this diagram. We see two maps; many

[13] Mechanical Turk website: www.mturk.com. For this task, we paid $0.01 for explaining the diagram with text. For each sentence, we collected responses from10 unique workers.

people might not understand which country or place is this. Even if they were to somehow interpret first one as Gaul and the second as Rome, they will read it wrong as *Gaul conquered Rome*, because it is linearly ordered, instead of using subject, event, object structure like ours. On the contrary, our diagram for the same example, failed to get a good representative picture for Rome and the Stanford parser failed to find that *1st century BC* is also related to the event ***conquered***, but with structure and text, many people were able to understand the content and produced something very similar to the original summary text.

Participants provided explanations of the diagrams (both those generated by our system and those of the two annotators) in English text from 10 different turkers for each sentence. We used Rouge [27], the automatic evaluation toolkit for summarization, to test how well their explanations retained the information of the original sentence's summary. We generate the reference summaries using annotators' identified entities and events and ordered them linearly like the diagram. For the example given above, our annotator's reference sentence summary was *"Gaul* ***conquered*** *Rome Julius Caesar 1st century BC"*.

These reference summary sentences are not grammatical and only consisted of the main event and the important entities. The Rouge evaluation handles this well because it is based on ngram matching and does not consider the grammaticality of sentences. For each system, we get the average Rouge score for each sentence (averaging over 10 turker's score) and then average over all sentences. We also average the two annotators' score and report the average annotator Rouge score.

In reporting our performance, we report both Rouge-1 and Rouge-L, since Rouge-1[14] and Rouge-L perform very well in evaluating very short summaries (head-line like summaries) [27]. In reporting our results, we are reporting precision (P), recall (R) and Fscore (F).

Evaluation	**Rouge-1**	**Rouge-L**
Explanation of Annotators' diagrams	**0.0892482 (F)**	**0.08451066 (F)**
	0.0680995 (R)	0.0635695 (R)
	0.1294495 (P)	0.1260265 (P)
Explanation of the ROC-MMS diagrams	**0.2405093 (F)**	**0.21649513 (F)**
	0.26668 (R)	0.23619 (R)
	0.2190162 (P)	0.199832 (P)

Table 5. Rouge-1 and Rouge-L for explanation of annotators diagram (average) and our system diagram

The results match our intuition that participants didn't do a very good job explaining the diagram with a sentence when they are provided with only pictures – even though human annotators selected these pictures. On the other hand, our system, despite the possibility of cascading errors from parsing, main event identification, entity extraction and identifying appropriate picture, did a lot better.

Although the inclusion of text gave the MMS diagrams a bit of an advantage in the Rouge score measurement because it is based on ngrams, it suggests that ROC-MMS is able to accurately identify the main concepts of the sentences and create pictures that are reasonable. More broadly, this evaluation shows the advantage of adding even minimal text, as many participants' were largely unable to produce accurate descriptions of the diagrams containing only pictures. Surprisingly, few participants simply wrote the text contained within the MMS diagrams, suggesting that the evaluation was more nuanced.

We believe that MMS diagrams will eventually be helpful for people who have trouble reading and understanding complex text and may help capable readers more easily skim documents. The end goal of MMS will be its ability to improve reading comprehension; ROC-MMS represents an important step in this direction.

FUTURE WORK

We evaluated ROC-MMS in the Wikipedia to show that multimodal summarization can be applied to complex text in order to generate diagrams that combine text, pictures, and structure. These evaluations have shown the promise of creating MMS diagrams completely automatically for arbitrary text, and suggest numerous future research opportunities.

First, our system currently relies partly on Wikipedia. An obvious extension would be to explore its performance in raw text, and adapt its modules to handle more general resources. The TRIPS parser used in ROC-MMS, already identifies named entities, which may be able to use to find better pictures for specific kind of entities, e.g., for people - we might search for portrait, for country – a flag or map.

Multimodal summarization is in the middle of two extremes. One would be to consider all events, instead of main events, i.e. represent everything with pictures and text. This may be useful for people who have trouble reading and want to get as much information in multimodal representation as possible. The other extreme is applying the summarization to pick the important sentences first and then apply multimodal summarization only on the selected sentences. In this way, it will represent the important sentences and only the important information in those sentences. This could be very useful for capable readers to skim through articles. Exploring the relative benefits along this dimension could better characterize their potential.

We simplified the problem of illustration by not representing events with pictures because events are usually hard to depict. Future work may try to illustrate events by more intelligently searching for events along with the

[14] Rouge-1 is based on unigram and Rouge-L is based on longest common subsequence.

subject and object. We also want to extend the proposed multimodal summarization by adding speech modality [15].

Finally, we want to extend our evaluation to look at how MMS (and other summary techniques) improve reading comprehension for the target groups who motivated this work – specifically people who have difficulty reading.

CONCLUSION

In this paper, we approached the problem of visualizing text as multimodal summarization. To create MMS diagrams, we automatically summarize text by extracting simple sentence structures (subject – who did it, event – what happened, object – to whom, preposition – how) and illustrate the text with pictures and compressed text together. Our evaluation showed that we achieve good performance on all of the subtasks required to create MMS diagrams, and that the MMS diagrams generated by ROC-MMS were easier to understand than human illustrations with pictures alone. Our implementation and evaluation leveraged the Wikipedia domain, but the approach embodied in ROC-MMS can be generally extended to unrestricted text.

ACKNOWLEDGMENT

We thank the three anonymous reviewers for their valuable feedback. We also thank Benjamin van Durme for his suggestion of prototyping on the Wikipedia domain, and Anna Loparev, Amal Fahad and Shantonu Hossain for help with annotation tasks.

REFERENCES

1. R. N. Carney and J. R. Levin, "Pictorial Illustrations Still Improve Students' Learning from Text," *Educational Psychology Review,* vol. 14, 2002.
2. B. Goldberg*, et al.*, "Easy as ABC? Facilitating pictorial communication via semantically enhanced layout.," *Twelfth International Conference on Computational Natural Language Learning*, 2008.
3. R. Mihalcea and B. Leong, "Toward communicating simple sentences using pictorial representations," presented at the Association of Machine Translation in the Americas., 2006.
4. J. Zhu, et al., "A text-to-picture synthesis system for augmenting communication.," in The Integrated Intelligence Track of the Twenty-Second AAAI Conference on Artificial Intelligence, 2007.
5. K. Barnard*, et al.*, "Matching words and pictures.," *Machine Learning Research,* vol. 3, pp. 1107–1135, 2003.
6. D. Joshi*, et al.*, "The story picturing engine—a system for automatic text illustration.," *ACM Transactions on Multimedia Computing, Communications, and Applications,* vol. 2(1), 2006.
7. Paivio, "Mental representations: A dual coding approach," *New York: Oxford University Press.,* 1986.
8. M. Glenberg, "Component-levels theory of the effects of spacing of repetitions on recall and recognition.," *Memory and Cognition,* vol. 7, pp. 95-112, 1979.
9. R. G. Greene, "Spacing effects in memory: Evidence for a two-process account.," *Journal of Experimental Psychology: Learning. Memory. and Cognition,* vol. 15, pp. 371-377, 1989.
10. M. Glenberg and W. E. Langston, "Comprehension of illustrated text: pictures help to build mental models.," *Memory and Language,* vol. 31, pp. 129–151, 1992.
11. R. E. Mayer, *Multimedia learning*. Cambridge, UK: Cambridge University Press., 2001.
12. U. Frith, "A developmental framework for developmental dyslexia," *Annals of Dyslexia,* vol. 36, pp. 69-81, 1985.
13. S. L. H. Association, "Roles and responsibilities of speech-language pathologists with respect to augmentative and alternative communication: Technical report," *ASHA Supplement,* vol. 24, 2004.
14. N. UzZaman*, et al.*, "Pictorial Temporal Structure of Documents to Help People who have Trouble Reading or Understanding. ," International Workshop on Design to Read, CHI, Atlanta, GA, 2010.
15. J. P. Bigham*, et al.*, "WebAnywhere: A Self-Voicing, Web-Browsing Web Application," International Conference on the World Wide Web, Beijing, China, 2008.
16. K. Knight and D. Marcu, "Summarization beyond sentence extraction: a probabilistic approach to sentence compression," *Artificial Intelligence,* vol. 139, pp. 91–107, 2002.
17. J. Pustejovsky*, et al.*, "TimeML: Robust Specication of Event and Temporal Expressions in Text. ," in *New Directions in Question Answering*, 2003.
18. J. Pustejovsky and M. Verhagen, "SemEval-2010 task 13: evaluating events, time expressions, and temporal relations (TempEval-2)," Workshop on Semantic Evaluations: Recent Achievements and Future Directions, 2010.
19. Y. Matsuo and M. Ishizuka, "Keyword Extraction from a Single Document Using Word Co-Occurrence Statistical Information," *International Journal on Artificial Intelligence Tools,* vol. 13, pp. 157-170, 2004.
20. R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, 2004.
21. R. Datta*, et al.*, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.,* vol. 40, pp. 1-60, 2008.
22. Coyne and R. Sproat, "WordsEye: An automatic text-to-scene conversion system," SIG-GRAPH, 2001.
23. K. Barnard and D. Forsyth, "Learning the Semantics of Words and Pictures," Eighth International Conference on Computer Vision (ICCV'01), 2001.
24. J. Lafferty*, et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," International Conference on Machine Learning, 2001.
25. N. UzZaman and J. F. Allen, "TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text," International Workshop on Semantic Evaluations, ACL 2010.
26. J. F. Allen*, et al.*, "Deep semantic analysis of text," Symposium on Semantics in Systems for Text Processing (STEP), 2008.
27. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," ACL Text Summarization Workshop, 2004.