A preprocessing system to include imaginative animations according to text in educational applications *

Eric Charton, Michel Gagnon, Benoit Ozell
École Polytechnique de Montreal
2900 boulevard Édouard-Montpetit, Montréal, QC H3T 1J4, Canada
{eric.charton|michel.gagnon|benoit.ozell}@polymtl.ca

Abstract

The GITAN project aims at providing a general engine to produce animations from text. Making use of computing technologies to improve the quality and reliability of services provided in educational context is one of the objective of this project. Many technological challenges must be solved in order to achieve such a project goal. In this paper, we present an investigation on limitation of text to graphics engines regarding imaginative sentences. We then comment preliminary results of an algorithm used to allow preprocessing of animation according to a text for an application software dedicated to multi-modal interactive language learning.

Keywords: Generation of animations

1 Introduction

In a long term perspective, The GITAN project[1] (Grammar for Interpretation of Text and ANimations), which started at the end of 2009, aims to solve the problem of transition from a textual content to a graphical representation. Discovering those mechanisms implies exploration of intermediate steps. As this project is generic and not domain dependent, we specifically need to explore the limits of computability of a graphic animation, regarding to a sentence, into a wide acceptance. In particular, we need to investigate the limits of existing graphic rendering techniques, regarding to the potential complexity of semantic meaning obtained trough a free, on the fly, sentence acquisition.

To illustrate this, we present preliminary results of a system dedicated to build a language learning software application. This system involves the capacity of a student to produce a semantically and syntactically acceptable sentence using a limited bag of words defined by a teacher, while observing a graphical animation of the sentence. The difficult aspect of this work is that the learning software have to display an animation for any syntactically correct sentence constructed from the bag of words. The idea is to allow the student to compare the animation that results from his own words arrangement with the one that conforms to the visual representation of the target sentence chosen by the teacher (see figure 1). An intuitive advantage of such a tool is the capacity given to the student to understand instantly, with the help of animations, misinterpretations and confusions resulting in some sentences constructions. Under a theoretical perspective, this application is an opportunity to investigate specific cases appearing in animation generation, driven by a non constrained natural language.

This paper is organized as follows. First, we describe the proposed application, and investigate the theoretical challenge arising from its specificities. Then, we describe the previous attempts made in the research field of text to animation systems, and put them into perspective with the specific problem encountered with open sentences generated from a bag of words. In the

*This work is granted by Unima Inc and Prompt Québec
[1] www.groupes.polymtl.ca/gitan/

fourth section, we present a system and its algorithms whose purpose is to anticipate the types of sentences that a student can produce from a bag of words and limitate the amount of animations to be preprocessed. Then we present the results of an experiment where we produce a delimited set of sentences extrapolated from a bag of words and evaluate how those sets can be used to pre-process animations. We conclude with future work.

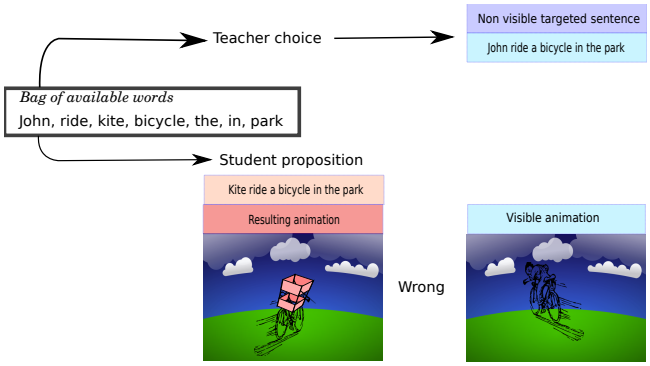

Figure 1: Synaptic representation of proposed application

2 Application principle and theorical view

Chomsky investigated one aspect of nonsensical meaning in sentence construction with its famous sentence *Colorless green ideas sleep furiously*[2]. This is an example of a sentence with correct grammar (logical form) but potential nonsensical content. Our application is a typical case of the need for acceptance and interpretation of potential nonsensical sentences. It has been shown by Pereira (2000) that such a sentence, with a suitably constrained statistical model, even a simple one, can meet Chomsky's particular challenge. Under this perspective, this can be view as a metaphoric problem, but not only: it can also deals with unnatural communication intent, relevant to pure imagination. This problem investigated by the linguistic theory as the transformation mecanisms of *conceptual-intention* into a linear sentence is not solved yet (Hauser et al. (2002); Jackendoff and Pinker (2005)).

In the generic field of graphical representation, Tversky et al. (2002) claim that correspondences between mental and graphical representations *suggest cognitive correspondences between mental spaces and real ones*. In the perspective of transforming a *conceptual-intention* into visual representation, Johnson-Laird (1998) consider that visual representation of mental models *cannot be reduced to prepositional representations*[3] *[as] both are high-level representations necessary to explain thinking*[4]. Johnson Laird consider also that *mental models themselves may contain elements that cannot be visualized*. According to this, it appears in the perspective of a text to animation computer application, that the correspondences between semantic abstractions extracted from free text and visual representations are not always relevant to a simple sentence parse and rendering in a graphic engine. In pictural arts, the correspondences for mental representations permitted by imagination, are obtained by a cognitive transformation of physical law, natural spaces and transgression of common sense to adapt an animation or a static image to the mental representation. Finally we can consider that animated results of those specific transformations are equivalent to the creative ones observed in artistic and entertainment applications like computer games, movies, cartoons. This particular aspect of natural language driven image generation and the role of physic limitations has been investigated by Adorni et al. (1984) who consider that such a cognitive transformation should be relevant to a computer IA problem.

[2]In *Syntactic Structures*, Mouton & Co, 1957.
[3]Defined by Johnson-Laird (1998), page 442 as *representations of propositions in a mental language*
[4]Johnson-Laird (1998) page 460

2.1 Three cases of syntactically correct nonsensical sentences

To illustrate this, let us consider a bag of words, including the 10 following terms: ***{Jack, rides, with, bicycle, park, the, kite, runs, in, his}***. According to the rules of our application, the learner is allowed to build any sentence including a subset of those words. Those sentences can be for example *Jack rides his bicycle in the park. The kite runs in the park.* But they can also be *The bicycle rides Jack. The kite rides the bicycle.* If we mentally imagine the scenes expressed by these four sentences, we intuitively know that each one can be animated. Some of them violate common sense or physical laws, but can still be animated. For example, we can produce an animation representing a *bicycle riding it's owner*, and thus revealing to the student a misinterpretation of relations between dependencies in a sentence. This is a **position case**. We will see that such semantic case can be represented by a graphic engine.

An other case could be a sentence based on **action** verbs. If we consider a bag of words containing ***{cat, eats, on, the, chair, in, his}***, a teacher will be able to define a target sentence like *The cat eats on the chair.* But the *eating* verb can have various possibles representations, according to the order of words, and can be organized in sentences like *The chair eats the cat. The chair eats on the cat.* Only a mental work can solve the problem posed by the visualization of these sentences, and this work imply attribution of an imaginative animation sequence describing a *chair eating.* We can imagine a metaphoric application using a classical graphic engine, where a *cat disappears when it is touching the chair.* But this is clearly a lack of realistic, difficult to accept with our education application.

A third case will involve **transformations**: if we consider now a bag of words containing ***{prince, transforms, into, the, castle, in, his, toad, himself, a}***. The targeted sentence could be *The prince transforms himself into a toad.* But it becomes difficult to integrate in a graphic engine a transformation function compatible with constructions like *The toad transforms him in into a Prince. The toad transforms the castle into a Prince.* If we consider all the possible action verbs and all the objects which can receive the faculty to do the concerned action, we obtain a very difficult problem to compute, relevant to an I.A system, like predicted by Adorni et al. (1984).

From previous examples, we can divide this representation problem in three family of cases: a **position case** (*The kite rides the bicycle*), an **action case** (*The chair eats the cat*) and a **transformation case** (*The toad transforms the castle into a Prince*).

3 Existing systems and previous work

Many experiments have been previously done in the field of text to animation processing. In this section we examine some of the previously described existing systems and investigate their capacities regarding our three text to animation semantic cases.

3.1 Capacities of existing animation engine

In Dupuy et al. (2001), a prototype of a system dedicated to visualization and animation of 3D scenes from car accident written reports written is described. The semantic analysis of the CarSim processing chain is an information extraction task that consists in filling a template corresponding to the formal accident description: the template constrained choices limitate the system to a very specific domain, with no possible implication in our application context.

Another system, WordsEye, is presented in Coyne and Sproat (2001). The goal of WordsEye is to provide a blank slate where the user can paint a picture with words: the description may consist not only of spatial relations, but also actions performed by objects in the scene. The graphic engine principle of WordsEye, like most of graphic engines, is able to treat the **position case** like *A chair is on the cat* [5] but because of its static nature, offers no possibilities to treat neither the **action cases** nor the **transformation cases**. Authors of WordsEye considers that *it is infeasible to fully capture the semantic content of language in graphics*[6].

[5] Numerous examples are available on the website at www.wordseye.com

[6] in Coyne and Sproat (2001) page 496

In academic context, the system e-Hon, presented by Sumi and Nagata (2006), uses animations to help children to understand a content. It provides storytelling in the form of animation and dialogue translated from original text. The text can be a free on-the-fly input from a user. This system operates in a closed semantic field[7] but uses an IA engine to try to solve most of the semantic cases. Authors indicate that some limitations have been applied: firstly, *articulations of animations are used only for verbs with clear actions*; secondly, this system constrains sentences *using commonsense knowledge in real time* (using ontological knowledge described in Liu and Singh (2004)). It is interesting, regarding our targeted application, to observe that a system dealing with potentially highly imaginative interactions from children needs to restrict its display with a *commonsense resource*.

Some applications like Confucius Ma (2006) are more ambitious. The animation engine of Confucius accepts a semantic representation and uses visual knowledge to generate 3D animations. This work includes an important study of visual semantics and ontology of eventive verbs. But this ontology is used to constrain the representation[8] to commonsense[9] trough a concept called *visual valency*. According to this, Confucius technics cannot fit with the studied cases of our application.

Finally, the main characteristics of most of those existing systems are that they operate in a closed semantic field, according to common sense and respecting physical laws. One of them (WordsEyes) can represent any spatial position for any object in a scene. But none of those existing systems has the capacity to produce realistic representation for usage of action verbs non conform to common sense included in a syntacticly correct sentence and none of them can manipulate a transformation of any concept to another. This establish a clear limitation of actual technologies available for the text to animation task when they are used in an open semantic field.

3.2 Semantic parsing and generation from bags

Besides, as discussed earlier, our application may meet situations where the animation does not respect physical laws and common sense. We have shown that there is many cases where it is not possible to simply parse an input sentence from the user and produce on the fly a semantic specification and give it to an animation engine. If grammar does not contain commonsense or physical laws, the semantic content of a syntacticly correct sentence can correspond to a mental representation that does not respect common sense and that is not compatible with any actual existing animation engine. According to this, in our application context, one possible way is to try enumerating all the possible sentences that a bag of words can generate and to see if there is a way to cluster those sentences of similar meaning into sets small enough to be compatible with a pre-processing animation task. This is a typical sentence realization task, actively investigated in Natural Language Generation (NLG) (see Reiter and Dale (2000)). Text generators using statistical models without consideration to semantics exists. Langkilde and Knight (1998) present a text generator would take on the responsibility of finding an appropriate linguistic realization for an underspecified semantic input. In Belz (2005), an alternative method for sentence realization very close to our needs uses language models to control formation of sentences. However, our problem is specific and difficult to solve with a NLG module as we need to produce all possibles sentences from a bag of word to preprocess animations, and not only a unique well formed sentence, corresponding to a *conceptual-intention*. This specific aspect of exhaustive generation from bag of words has been first investigated by Yngve (1961). In this work, a generative grammar is combined to a combinatorial random sentence generator applied to a bag of words. Most of the output sentences were quite grammatical, though nonsensical. Recently, Gali and Venkatapathy (2009) explored a derived work where models consider a bag of words with unlabeled dependency relations as input and apply simple n-gram language modeling techniques to get a well-formed sentence.

[7] 18 characters, 67 behaviors, and 31 backgrounds

[8] Ma (2006) page 109

[9] *Language visualization requires lexicalcommon sense knowledge such as default instruments (or themes) of action verbs, functional information and usage of nouns.* Ma (2006) page 116

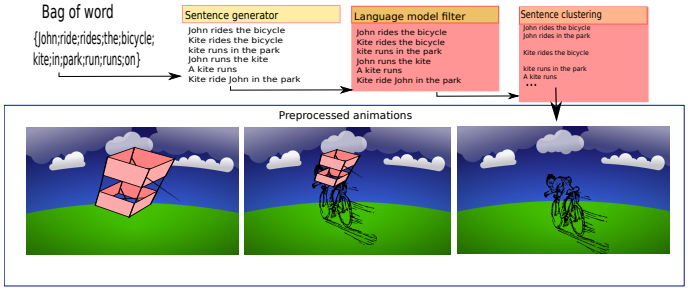

Figure 2: Architecture of the system and its successive algorithms

4 Proposed system

The given problem could be solved through enumeration of all the syntacticly valid sentences that may potentially be produced for a given bag of words, without consideration to semantics, common sense or physical laws, followed by a clustering of those sentences into groups according to their meaning similarity. First, our system takes as input a bag of words and produces all syntacticly valid sentences by means of a simple English rule-based sentence generator. Then, it uses a language model (as described in Song and Croft (1999)) to select, among the group of word combinations, only sentences that are valid according to a modeled language. Finally, a clustering algorithm groups these sentences by using a meaning similarity measure. At the end, we obtain for a given bag of words a restricted list of sentences, clustered by senses. We can produce for each cluster of sentences an unique animation. This unique animation will be displayed when the student makes an attempt of sentence construction.

4.1 Sentence generator (SG)

The sentence generator (SG) is built with a limited set of flexible generative grammar rules implemented in Prolog. Those rules, which cover verbal phrases, noun phrases and prepositional phrases, allow the generation of sentences from a bag of words. The category of the words contained in the bag is also considered and added as a label to each word contained in the generated sentence. For example, the rules for verb phrases are the following ones:

```
vp(Features,BagIn,BagOut)-->
        lex(v,Features,BagIn,BagOut).

vp(Features,BagIn,BagOut,)-->
        lex(v,Features,BagIn,Bag1),
        np(_,Bag1,BagOut,).

vp(Features,BagIn,BagOut,)-->
        lex(v,Features,BagIn,Bag1),
        pp(Bag1,BagOut).

vp(Features,BagIn,BagOut)-->
        lex(v,Features,BagIn,Bag1),
        np(_,Bag1,Bag2),
        pp(Bag2,BagOutt).
```

Note that the `lex` predicate refers to the lexical entry that specifies whereas `np` and `pp` refer respectively to noun phrase and preposition phrase rules that will be recursively applied. We can see that the verb phrase rules cover about all verb arities without constraints. As we will see later, it is the language model that will constrain the generative expressivity. The rules also take as parameters the bag of words and the sequence of words forming the sentence currently generated. At each step in the execution of a rule, words are extracted from the bag of words and appended at the end of the sequence.

The used word categories are described by a standard morphosyntactic tag from Penn-Tree bank tag-set[10] like noun (NN), proper name (NP), verb (VBZ), conjunction (IN), article (DT), personal pronoun (PP). SG generates a sentence by combining phrases. For example, a sentence can be produced by combining a verb phrase with a noun phrase at subject position, as expressed by the following grammar rule (note that there are agreement constraints for person and number, and another constraint specifying that the verb phrase must be in declarative mode):

```
s(BagIn,BagOut,SeqIn,SeqOut)-->
       np(pers~P..number~N,BagIn,Bag1,SeqIn,Seq1),
       vp(mode~dec..pers~P..number~N,Bag1,BagOut,Seq1,SeqOut).
```

Taking as input the bag of words {*the,is,a,Jack,bicycle,kite,park,in,rides,runs*}, the system generates the following sentences:

```
Jack/NP rides/VBZ the/DT bicycle/NN
Jack/NP runs/VB the/DT bicycle/NN
Jack/NP runs/VB the/DT kite/NN
the/DT bicycle/NN rides/VBZ Jack/NP
the/DT bicycle/NN rides/VBZ a/DT kite/NN
the/DT bicycle/NN runs/VB Jack/NP
...
```

The flexibility of this very simple generative grammar is a deliberate choice to avoid the risk of non-generation of a valid sentence. In case of a non-valid sentence, the next module of our system is a language model filter that has been trained with a big corpus and achieves a final filtering that will remove all non-valid sentences.

4.2 Language model Filter (LMF)

The language model (LM) is trained from a corpus which domain is related to the targeted application. For the sample application presented in this paper (teaching English language), we used the *Simple Wikipedia* corpus[11]. This corpus uses simple English lexicon and grammar and is well-suited for our application needs. The language model is trained with the SRILM toolkit. Each sentence proposed by the *Sentence Generator* is filtered by using an estimation of its probability, regarding LM. In our application, SRILM produces N-Gram language models of words[12]. With such a model, the probability $P(w_1, \ldots, w_n)$ to observe a sentence composed of words $w_1....w_n$ in the modeled corpus is estimated by the product of probabilities of the individual appearance of words contained in sequence $P(w_{1,n}) \approx P(w_1)P(w_2)...P(w_n)$. To obtain a more robust system, bi-Gram or tri-Gram models applied to a sequence of n words are adopted: $P(w_1, \ldots, w_n) \approx P(w_1)P(w_2|w_1)P(w_3|w_{1,2})...P(w_n|w_{n-2,n-1})$. In our application, we use a bi-Gram model, which can be represented by the following example:

$P(Jack, rides, the, bicycle) \approx P(Jack| < s >)P(rides|Jack)P(the|rides)P(bicycle|the)$

For each sentence generated by SG, we estimate its probability of appearance. The non-existence of a bi-Gram sequence means a null probability for the complete observation sequence and rejection of generated sentence. It is also possible to define a threshold constant to reject sentences with low probability estimation.

4.3 Clustering algorithm (CA)

The clustering algorithm uses the chunking faculty of the Tree-tagger morphosyntactic shallow parser[13]. Chunking is an analysis of a sentence that identifies the constituents (noun phrases, verb phrases, etc.), but does not specify neither their internal structure, nor their role in the main sentence.

[10] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/Penn-Treebank-Tagset.pdf

[11] See simple.wikipedia.org, and downloadable version on http://download.wikipedia.org/simplewiki/

[11] Available on http://www.speech.sri.com/projects/srilm/

[12] An n-gram is a subsequence of n items from a given sequence. The items can be phonemes, syllables, letters, words or base pairs, according to the application.

[13] The Tree-tagger is a tool for annotating text with part-of-speech and lemma information. It can also be used as a chunker. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Considering the list l of n sentences $1...n$ kept by LMF, we generate a function $f\_similarity$ for the first sentence $s1$ of l. This function contains a description of the nature of each phrase chunk and its position in $s1$. Each phrase chunk is associated with its lexical content, with consideration to similarities (i.e. two similar verbs will be considered as unique). Next, we submit sentences $2...n$ to $f\_similarity$, and group those which function has returned a 1 value. Finally we remove all the clustered sentences from l and iterate CA until l is empty. For the example *[Jack/NC] [rides/VC] [the bicycle/NC]* the similarity clustering function will be:

```
f_similarity(sentence) = {
  if (sentence={1:NC{Jack};2:VC{rides;run};3:NC{bicycle}})) return(1)
else return(0) }
```

And clustering will be :

```
[Jack/NC] [rides/VC] [a bicycle/NC]
[Jack/NC] [runs/VC] [the bicycle/NC]
[Jack/NC] [rides/VC] [the bicycle/NC]
```

5 Experiments and preliminary results

In the preliminary experiments of our system, we used 10 bags of 10 words. Bags of words come from exercises included in an learning English student's book[14]. Those exercises include,for a given topic, (i.e. *Talking about abilities*) a set of target sentences, and a suggested vocabulary (i.e.*play,guitar,dance,swim,etc*).

Words	Generated sentences (SG)	Correct sentences (LMF)	Sentences clusters (CA)
6	25	23	7
10	460	280	20

Table 1: Evaluation of group of sentences generated from a bag of words

We use 6 and 10 words from the bag and apply SG, LMF and CA. We count sentences generated in SG, kept in LMF, and how many clusters remain in CA. Table 1 gives the arithmetic mean value of results for each step of the test. This preliminary experiment confirms that for a given bag of words, it is possible to generate a limited set of semantics groups, compatible with a not expensive video preprocessing task. With a bag of 10 words, only 20 clusters are obtained, meaning only 20 animations have to be produced based on the limited set of objects delimited by the bag of words.

Those preliminary results are sufficient to build an application prototype. With such results, our system can be used to preprocess and help to evaluate amount and specificity of potential animations according to a bag of words used to produce sentences. Our method allows to select, for a given bag of words, a limited set of semantic groups of sentences. The system can be used as a production tool to preprocess video in a text-to-animation multimodal application. It can also be used as a component of text-to-animation application software to evaluate its semantic field and produces automatically test sentences for evaluation purposes.

6 Conclusion

We presented an original component to support text to animation applications. The originality of this system is that it is not restricted to valid semantic productions that do not violate common sense and physical laws. This proposition investigates the specific situation of imaginative text to image applications. We showed that a generative grammar combined with statistical methods can extract a limited amount of potential sentences from a given bag of words. The advantage of such a structure is its ability to preprocess text to animation sequences in an open context application, with a low amount of miss-representations of animated sequences regarding to text sense. The next step of our work is to try to introduce in our architecture a real-time text to image generator that accepts, in restricted semantic domains, scenes that do not respect common sense. This will be an attempt to evaluate the capacities of a system to elaborate imaginative-like text to animation system.

[14] *Go For It! English for Chinese students*, serie published by Thomson Learning.

References

Adorni, G., Di Manzo, M., and Giunchiglia, F. (1984). Natural language driven image generation. In *Proceedings of COLING*, volume 84, page 495500. 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics.

Belz, A. (2005). Statistical generation: Three methods compared and evaluated. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG05)*, volume 05pages, page 1523.

Coyne, B. and Sproat, R. (2001). Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, volume ternetchap, page 487496. ACM New York, NY, USA.

Dupuy, S., Egges, A., Legendre, V., and Nugues, P. (2001). Generating a 3D simulation of a car accident from a written description in natural language. *Proceedings of the workshop on Temporal and spatial information processing* -, pages 1–8.

Gali, K. and Venkatapathy, S. (2009). Sentence Realisation from Bag of Words with dependency constraints. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, number June, page 1924. Association for Computational Linguistics.

Hauser, M., Chomsky, N., and Fitch, W. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569.

Jackendoff, R. and Pinker, S. (2005). The nature of the language faculty and its implications for evolution of language (Reply to Fitch, Hauser, and Chomsky). *Cognition*, 97(2):211–225.

Johnson-Laird, P. (1998). Imagery, visualization, and thinking. *Perception and Cognition at Century's End*, pages 441–467.

Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, page 704, Morristown, NJ, USA. Association for Computational Linguistics.

Liu, H. and Singh, P. (2004). Commonsense reasoning in and over natural language. In *Knowledge-Based Intelligent Information and Engineering Systems*, page 293306. Springer.

Ma, M. (2006). *Automatic conversion of natural language to 3D animation.* Thesis, University of Ulster, Faculty of Engineering.

Pereira, F. (2000). Formal Grammar and Information Theory: Together Again? *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 358(1769):1239 – 1253.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems.* Cambridge University. Press.

Song, F. and Croft, W. B. (1999). A General Language Model for Information Retrieval. *Information Retrieval*, pages 316–321.

Sumi, K. and Nagata, M. (2006). Animated storytelling system via text. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, volume Proceeding. ACM SIGCHI international conference on Advances in computer entertainment technology.

Tversky, B., Morrison, J., and Betrancourt, M. (2002). Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247262.

Yngve, V. (1961). *Random generation of English sentences.* Number September. Massachusetts Inst. of Technology.