TOWARDS COMPUTER-GENERATED MNEMONIC PHRASES: EXPERIMENTS WITH GENETIC ALGORITHMS AND N-GRAMS

James Mountstephens
School of Engineering and Information Technology Universiti Malaysia Sabah Jalan, Kota Kinabalu, Sabah, Malaysia
E-Mail: james@ums.edu.my

ABSTRACT

Mnemonic phrases have the potential to help people commit information to memory and may be a valuable aid to education. However, their widespread application is currently limited by the effort and creativity required to generate them manually. This paper describes a method for the automatic generation of effective mnemonics by computer, framing the task as an optimisation problem to be solved by Genetic Algorithms using parser output and n-gram frequencies to evaluate fitness. Grammatical constraints and lexical familiarity are parameters tested for their ability to produce more memorable sentences. The method has been implemented using custom code and existing libraries, and tested, showing promising results on list data of increasing difficulty.

Keywords: mnemonics, genetic algorithms, N-grams, natural language, education.

INTRODUCTION

Memorisation is an integral part of learning and, although the development of thinking skills and creativity are equally important, successful education depends on the transfer, long-term retention and recall of large bodies of information and knowledge (Ellis-Ormrod, 2011; Worthen and Hunt 2010; Searleman and Herrmann, 1994). This material may be expressed in different modalities ranging from the symbolic and conceptual, which are largely presented in text form, to the spatial and relational which are often presented graphically (Ellis-Ormrod, 2011; Travers, 1982).

Lists are a form of unstructured text that may be especially hard to memorise since their items often have no obvious relationship to each other and no formula usually exists to generate them from first principles or from each other. Examples of lists in computing and engineering that a student must commit to memory might include the 7 layers in the OSI model of networking (Forouzan, 2013), colour coding for resistors (Yadav, 2005), the set of organic elements (Anslyn and Dougherty, 2006) and the activity sequence of metals (Jolly, 1991).

In the context of human learning and memory, mnemonics are "cognitive strategies designed to enhance the encoding of information" in order to facilitate storage and retrieval (Worthen and Hunt 2010: 2). Such memory aids have been in use for millenia and a wide range of techniques exist that are suitable for different types of information. Mnemonics for lists are most often 'first letter mnemonics' that perform a tranformative encoding (ibid.) of the list based on its initials. The computing world is replete with *acronyms*, where the list initials spell a pronounceable (if not always meaningful) word. Elementary examples include: FIFO, RAM, RAID. Acronyms work well when the list initials are pronouncable but this is not always the case. More generally, mnemonic phrases can be used. The most common type of mnemonic phrase is a sentence with the same initials and number of words as those in the list to be memorised. Classic examples for some common lists are:

a) 'My Very Excellent Mother Just Sent Us Nachos' for the eight planets Mercury, Venus, Earth, Mars,Jupiter, Saturn, Uranus, Neptune.
b) 'Richard of York Gave Battle in Vain' for the seven colours Red, Orange, Yellow, Green, Indigo, Blue, Violet.
c) Every Good Boy Deserves Food' for the notes on the musical stave E, G, B, D, F.

Research in education and psychology indicates that, used properly, mnemonics can genuinely aid memory and learning (Seay, 2010; O'Hara 2007, Glynn, Koballa and Coleman, 2003; Levin and Nordwall, 1992) but also that their general application can be limited by the time and effort required to generate them. Currently, students must either use existing 'classic' mnemonics such as the preceding, which are of course limited in number, or devise their own. Self-created mnemonics have been shown to be effective but the task is challenging since imagination and the ability to satisfy the constraints of fixed sentence length and initials are required.

No known work has been done on the automatic generation of mnemonic phrases by computer and the early stages of work towards this problem are addressed in this paper. Effective, computer-generated mnemonics could aid the learning process of many students, regardless of discipline, and might also be extended to cover other types of knowledge.

EFFECTIVE MNEMONICS

A key principle of effective mnemonics is the association of to-be-learned information with something vivid and meaningful, which may be a factual or imagined, making it more structured and easily memorable (Worthen and Hunt 2010; Searleman and Herrmann, 1994; Travers, 1982). This is true of mnemonic phrases for lists, where the lack of obviously memorable structure is compensated for by the properties of the meaningful sentence it is associated with. While a list may be unstructured and without formulaic relation between elements that would allow deduction of the next item, a sentence is intimately bound together with long-range grammatical and semantic dependencies that constrain the possible options. The overall meaning of the sentence acts as a summary to both contain and bind its constituents, especially if it can produce vivid mental imagery.

Consider these principles in relation to the classic mnemonic phrases of the previous section. The fact that they are genuine sentences and not unrelated items is crucial. For our purposes, this sentencehood means that the words form a gramatically-acceptable structure in terms of phrases and atomic parts-of-speech (POS) such as nouns, verbs and adjectives. But, more than just syntax, meaning makes the phrases work. Each of the examples asserts something meaningful and capable of triggering or creating a vivid or familiar mental image. Richard of York really did give battle in vain; mothers frequently do give their children nachos; and, all other things being equal, good boys do deserve food. This meaning may be understood at both a general and specific level. In general terms, the entities in these phrases are *compatible*; the subjects perform believable verbs on believable objects. If My Very Excellent Mother Just Sailed Under Niagara, we would likely be surprised. The meaning may also go further to assert a *specific fact* rather than a generality, as in the case of Richard of York.

Whether well-known or general entities should be used in mnemonic phrases is not clear. Psycholinguistic research suggests that more general concepts are often harder to retrieve and/or picture and conversely that familiar entity may activate a schema of related concepts more readily (Miller, 1991; Gilhooly and Logie, 1980). Such related concepts may offer useful ammunition for establishing coherence in a phrase. For the example phrases, the situation is not clear cut. Certainly, Richard of York is a famous person in western culture, and those who know of him might be expected to think of battles and of losing. 'My very excellent mother' is unspecified by name but should be easily picturable to most people. But 'every good boy' seems less easily picturable. To explore the question, this distinction between well-known and general entities will be a key parameter in the experiments described later.

A related question is that of bizarreness. The example phrases are arguably successful because they make true, familiar statements but the history of mnemonics suggests that bizarreneness can be used effectively memorable (Worthen and Hunt 2010; Searleman and Herrmann, 1994). 'Richard of York Grabs Bears in Venice' is also capable of inspiring a vivid picture but our initial experiments have shown that bizarreness is hard to get right and can easily descend into gibberish. For example, 'Richard of York Gets Bravely into Vines' somehow does not cut it. Investigating the use of bizarreness will be left for future work.

With these concerns for grammar and meaning in mind, the problem may now be formalised.

GENERATING MNEMONIC PHRASES

For an input list L of n words with initials I_L, the output mnemonic phrase m_L will also be a sequence of n words with initials I_L. The question is: which sequence of n words will best satisfy the conditions of grammar and meaning just discussed? In this paper, the task will be framed as an optimisation problem to be solved by search, with a search space defined as follows.

Let W be a lexicon (a master list of words) and $L \in W^n$ be a list of words w_L to be remembered. L is of length n and has initials I_L. Let $m_L \in W^n$ be a possible mnemonic phrase for L. m_L is a sequence of words w_{mL} also with length n and initials I_L but $w_{mL} \neq w_L$. Let M_L be the set of all m_L determined by W and L. M_L is the search space for this problem. It is important to realise that W determines the expressivity of the possible sequences, and, since L is of fixed length, the size of W determines the size of the search space. It should also be clear that the vast majority of sequences in M_L will be neither grammatical, meaningful sentences nor useful mnemonics; they might be expected to have no more memorable structure or content than the original list L. The challenge will be to select the best available sequence $m_L \in M_L$ and we therefore need to define an objective function to reflect our criteria of grammar and meaning. This function will be of the form

$$f(m_L) = \mathrm{gr}(m_L) + \mathrm{mn}(m_L).$$

Grammar

The function $\mathrm{gr}(m_L)$ must evaluate the grammaticality of candidate word sequences and, since the purpose of natural language parsers is determine this structure in terms of phrases and parts-of-speech (POS), their use will be explored.

Parsing is one of the oldest topics in computational linguistics and there are many types of parsers which vary in appropriateness for the task at hand. Older parsers based on precision grammars will simply reject any sentences that do not fit their grammar (Wagner, Foster and Genabith, 2007; Chomsky, 1957). Although this decisiveness is attractive, its cost in expressiveness may be too high. Strict parsers are not well suited to dealing with the vast number of looser grammatical structures that people use everyday and some linguists believe that making sentencehood as black and white as this is not useful since people are usually capable of imposing some structure on even unusual sequences of words. Modern parsers based on Probabalistic Context-Free Grammars (PCFGs) (Klein and Manning, 2003) or Link Grammars (Sleator and Temperly, 1995) can handle more possible grammatical structures and are designed to be tolerant of some level of grammatical abberation. They can thus handle far larger domains but at the price of accepting some sentences that humans would consider ungrammatical.

For the purposes of this paper, use of modern, more tolerant parsers will be explored since the nature of the task already imposes strict constraints of sentence length and initials and so, for certain lists, a slightly more flexible grammatical structure may be needed to make any kind of sentence at all. This flexibility may be controlled with a novel means of imposing grammatical strictness that will be described later. For the function gr(m_L), modern parsers usually have a confidence level or cost function accompanying their parse tree and that will be used for evaluating grammaticality.

Meaning

Building a function mn (m_L) to evaluate a sentence for the criteria of meaning described earlier is highly challenging. To reward sentences where the content is compatible requires general knowledge of associations between objects, their properties and possible actions. To direct sentence content towards the assertion of specific facts like those concerning Richard of York potentially requires autobiographical and historic knowledge of the specific actors involved.

A knowledge base may be explicitly stated or may be implied by statistical regularities in appropriate data (Russel and Norvig, 2010). Explicit knowledge requires contruction and the development of suitable inference strategies, which are beyond the scope of this initial work. Here, implicit knowledge is explored in the form of n-gram frequencies derived from large bodies of machine-readable text known as corpora (Biber, Conrad and Reppen, 1998; Sinclair, 1991). N-grams are contiguous sequences of n items in a text and may be used to evaluate meaning implicitly. Example bigrams (when n=2) might include 'at home' and 'apple hermit'; and trigrams (when n=3) might be 'on the road' and 'octopus trait rheumatism'. It should seem reasonable that the first example of each n-gram would have a higher frequency than the latter and this information can be used to evaluate candidate sentences. Those with higher frequency should produce higher values of mn(m_L).

Although not explicitly, n-grams embody some grammatical and semantic regularities, as well as familiarity of usage. Words co-occurring are likely to be semantically related and the longer the n-gram, the longer the range of the dependency between objects in a sentence. Although long range dependencies are desirable, the use of the n-grams of the same length as the input list to be evaluated would be problematic since the frequency of any particular n-gram decreases rapidly as n increases (Biber, Conrad & Reppen, 1998). Matching and counting pairs of words will produce plentiful results but matching and counting a specific long sequence will not. The strictness of longer n-grams means they can provide no frequency for the vast majority of candidate word sequences. This strictness means that local coherence is also not rewarded. For example, when n=5, neither of the sequences 'cat hat you green backwards' or 'people like hurriedly high mountain' are likely to be found in a corpus but the latter sequence is more locally coherent than the first due to pairs of frequently-co-occuring words. Even if this were not an issue and strictness is acceptable, n-gram frequency decrease as n increases placing practical limits on n-gram sizes since the corpus size must increase dramatically to achieve statistically significant frequencies. For current corpora, n is usually no greater than 5. To evaluate sentences longer than five words, a simple frequency count is not possible.

For these reasons, the way of using n-grams here to evaluate meaning is to check each possible n-gram (n=2,3,4,5) on each possible word in the candidate sequence from left to right, taking care to not exceed the right end of the sequence. Using n-grams of all sizes will rewards pockets of local coherence even when the whole sequence does not match the contents of the corpus.

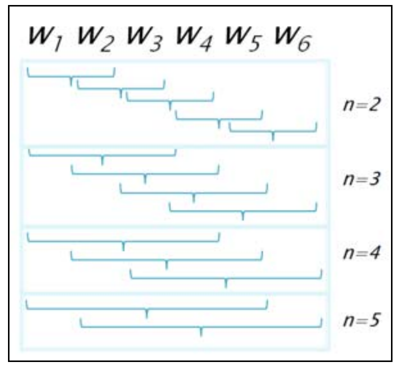

Figure-1. Summing all possible n-grams in all possible positions.

Genetic Algorithms

Given the components of the objective function just described, a search method must be chosen. Exhaustive search is arguably the best but the search space is likely to be large for a lexicon of reasonable size. For this research, Genetic Algorithms (GA) will be used to perform the search since they have proven to be robust and capable of rapid convergence in a wide range of problems (Russel and Norvig, 2010; Mitchell, 1996).

Taking inspiration from biological reproduction and evolution, the basic unit in a GA is the chromosome, which comprises a sequence of genes taking particular values known as alleles. For the generation of a mnemonic phrase for the organic elements (carbon, hydrogen, nitrogen, oxygen, phosphorus, sulphur), example chromosomes might be as below:

C1	[***cut his nose*** on plateau spike]	f=0.4
C2	[curt house nine over plague socks]	f=0.1
C3	[couple house never ***off peter smith***]	f=0.4
C4	[cup hat naughty on peter sausage]	f=0.1

Figure-2. Four example chromosomes, each with with six genes.

Although the specifics of GAs are beyond the scope of this paper, the essential idea is that initial populations of chromosomes are assessed for fitness (the GA version of an objective function) and the fittest

VOL. 9, NO. 9, SEPTEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

allowed reproducing to form new chromosomes. In the example above, the left and right halves of C1 and C3 have high fitness and might combine to form the better chromosome [cut his nose off peter smith] which is a grammatical sentence and a reasonable mnemonic phrase for the list of organic elements. The process of evolution is continued until chromosomes with a desired level of fitness are produced.

Extending the Search

A considerable number of options exist to extend the basic search framework just described. Two key parameters that will be tested are i) the effectiveness of the more flexible parser chosen earlier in achieving correct grammar and ii) whether well-known or general entities give the best results. Both questions can be addressed by placing appropriate constraints on the search and determining which works best.

Constraining Grammar

The tradeoff between strict and flexible grammar has been noted and it is understood that the choice here of a flexible parser may not be appropriate; it may be necessary to impose stricter grammatical constraints on word sequences generated. One method, explored here, is to determine a legal grammatical structure before the search begins, effectively binding positions in all candidate sequences to particular parts-of-speech. The search will still continue using the parser and n-grams but the output can be expected to be at least an acceptable sentence. This constraint may also vastly reduce the search space since, at a given position in the candidate sequence; allowable words must not only have the correct initial but must also have the correct POS.

Phrase Structure Grammars (Russel and Norvig 2013, Sag and Wasow 1999; Chomsky 1957) are explored for this task. PSGs are essentially rewrite rules for parts-of-speech, as illustrated below.

S	→	NP VP
NP	→	(art) (aj) n (PP)
VP	→	v (NP) (PP)
PP	→	p NP

Figure-3. Simple Phrase Structure Grammar (PSG).

These rules may be used to recursively divide the input list L into phrases and ultimately into atomic POSs like nouns and verbs. If the assignment is possible, it is guaranteed to be a grammatical sentence. The added difficulty in this context is that the words in the sequence are bound to fixed initials and the assigned POS must exist for a given initial. An example of this process successfully applied to L= {E, G, B, D, F} using the PSG from Figure-3 is shown below.

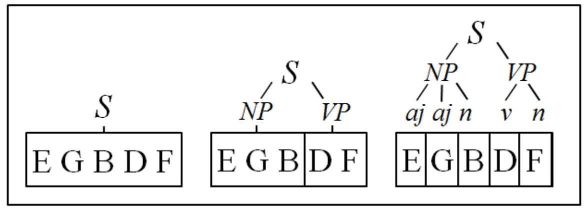

Figure-4. Success in constraining grammar using the PSG in Figure-3.

The division of the word sequence may vary. In this example, the initial split could have been NP = {E, G, B} and VP = {D, F} instead. Different divisions will yield different sentence structures and in certain cases will produce a sentence structure for which words with the correct initial for the assigned POS do not exist. An example of this situation might occur for L= {carbon, hydrogen, nitrogen, oxygen, phosphorus, sulphur} using the same PSG.

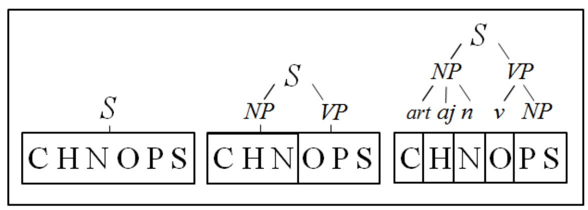

Figure-5. Failure in constraining grammar using the PSG in Figure-3.

At the third stage, the initial 'C' has been assigned the POS 'article' but there is no English article beginning with 'C', rendering the sentence unrealisable. Steps must be taken to check if a generated sentence structure is compatible with the given initials and if not, to redivide the sequence and recurse again, which will assign a different POS to the initial in question, as shown below.

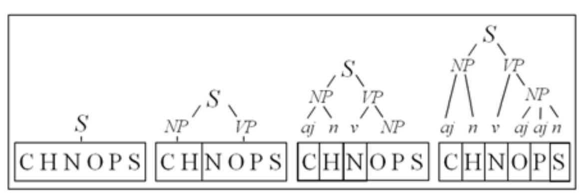

Figure-6. Correction of invalid assignment by redividing sequence.

Constraining Lexicon

Earlier, the issue of the specificity of the entities in the sentence was raised. It was asked whether sentences about famous people and the entities associated with them are intrinsically more vivid and memorable than sentences about general nouns such as people and cars.

To explore this in experiment, famous names and associated entities may be used simply by binding the names to positions in the sequence before the search begins. In essence, leaving gaps to be filled in with more general words. This may combined with the grammatical constraints described in the previous section to also bind the POS of the remaining placeholders.

To automate this process, the input list L is scanned for locations to insert famous names and their associated entities based on compatibility of initials. So for CHNOPS, the famous person Cleopatra might be assigned to the initial C, making her the subject of the sentence. Cleopatra is associated with a number of other entities such as hieroglyphics, needles, the Nile, Osiris, pyramids, pharoahs, sarchophagi and the Sphinx, that could potentially make a coherent sentence and have initials compatible with other positions in the list. It should also be clear, however, that there is only room for one or two of these other entities in the short sequence of six words and that space must be left for verbs, adjectives and function words to make a genuine sentence.

IMPLEMENTATION

The method described so far is intentionally general and a number of parameters must set concretely in any implementation of it. The implementation choices used in the experiments described later are described next. In addition to the use of existing libraries, a considerable body of custom Java code was developed for reading and converting linguistic data, interfacing GA and parser libraries and implementing the grammar and famous names constraints described in section 3.

Lexical Data

Choice of lexicon W is crucial since it determines the content, coherence and expressiveness of the phrases that can be generated. It also determines the size of the search space for the problem and should therefore not be too large. A number of options were considered for both general words and famous names.

At the extreme end of expressiveness for general words, we might use the set of distinct words in a large natural language corpus such as the British National Corpus (BNC, 100 million words) (Burnage, 1991) or the Corpus of Contemporary English (COCA, 450 million words) (Davies, 2011) since they both have lexica of over 500,000 distinct words and it would also be convenient to extract n-grams from the same data. But, even if efficiency were not an issue for such a large lexicon, our initial experiments suggested that the vast majority of sequences generated were uncomfortably odd since words from many different contexts were being combined. Certainly, frequency could be used to determine the most frequent and reject the rest but even if frequency was able to determine a smaller set of more familiar words there is still no guarantee that they form an expressive and coherent set. Inspection of the BNC suggests not and it will be left for future work to explore different methods of using the lexica of corpora directly.

Alternatively, specialised languages are designed to be coherent and expressive using a small number of words. Ogden's Basic English (BE, 850 words) (Ogden, 1937) was a major candidate and was used in early tests. However, BE does not directly give POS categories to its words and expressly tries to eliminate verbs in favour of a small number of action words, resulting in sentences that sound extremely unnatural. Another specialised language, The Voice of America (VOA) 1500 word list (Voice of America, 2013), can be considered a simplified, distilled version of English intended for description since it was devised to allow broadcast of the same content in 46 different language. Unlike BE, it has verbs and six simple POS tags. This is the lexicon used here.

A number of sources of famous names were also considered and judged based on expected quality and ease of machine extraction. The Notable Names Database (NNDB) (NNDB, 2013) features a large number of famous people and the accompanying textual biographies contain the potential for extracting terms associated with these people. As an online-only database, it would require considerable work to extract the basic data and, since the biographies are unstructured text, to extract genuine associated terms.

Instead, the results of an online poll attempting to decide the most famous people (Famous, 2013) was used as a basic list of 150 famous people and included such as Marilyn Monroe and Winston Churchill. 20 associated terms for each name were derived using Google Sets, whose purpose is to do exactly that. It should be noted that Google Sets has unfortunately been closed down and its functionality is now only accessible manually through Google Sheets, an issue we hope to automate soon.

N-Gram data

N-gram data was taken from the 450-million word Corpus of Contemporary American English (COCA). The top million n-grams for n=2, 3, 4, 5 are available free in a convenient text file format (N-grams, 2013). The lexicon determined by the n-grams is far larger than the VOA chosen for these experiments and was therefore aligned and substantially trimmed by mapping across lemmas and differing POS conventions. The separate lists for each n were combined into a single tree structure for more efficient querying. Although we would have liked to use British English, COCA is 4.5 times bigger than the BNC and its size allows better frequency estimates for longer n-grams making it the best choice here.

GA Library

The Java Genetic Algorithms Library JGAP (JGAP, 2013) is a freely-available implementation of many established genetic algorithms. It offers classes and methods for generating and evolving populations and calculating customised fitness functions.

Parser Library

Two main options for parser library were considered in this implementation. Arguably the most popular freely-available general english parser, the Stanford Parser (Stanford, 2013) is considered state-of-the-art and can generate a parse probability that can be

VOL. 9, NO. 9, SEPTEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

used in the fitness function required here. Alternatively. The CMU Link Parser (CMU, 2013) is freely-available as a Java library and although poorly-documented, it can be used to generate linkage and cost data for a given sentence.

The choice was made in favour of the Link Parser after initial experiments demonstrated that the Stanford parser was considerably slower (about 2 secs vs 0.5 secs per word sequence). Note that the Link Parser has a cost function rather than a parse probability and higher values will therefore reduce the fitness function rather than increase it.

EXPERIMENTS

Since our method is still being refined, evaluation of the mnemonics generated will be by inspection here. With more refinement, the mnemonics will soon be tested on real students and their effectiveness will be determined more objectively. Four lists of increasing length were used:

a) Undergraduate computer scientists learn that executable code is generated by ‘lexical analysis, parsing, code generation’. This list has five words and initials L A P C G.
b) The list of organic elements ‘carbon, hydrogen, nitrogen, oxygen, phosphrous, sulphur’. This list has six words and initials C H N O P S.
c) The seven layers in the OSI networking model, ‘physical, datalink, network, transport, session, presentation, application’ have initials P D N T S P A.
d) Electrical engineers need to memorise the list of 12 resistor colour codes: ‘black, brown, red, orange, yellow, green, blue, violet, grey, white, gold, silver’. Initials are B B R O Y G B V G W G S. This is the longest list and is expected to be challenging.

The experimental design is straightforward: after a randomly-generated baseline sequence is found for each list, the two parameters described earlier are varied: ie general nouns vs famous names and POS constraints vs no POS constraints. The results are described next.

Random Baseline

Unsurprisingly, random sequences of words chosen from even this relatively small and controlled lexicon do not yield meaningful grammatical sentences, as shown below.

Table-1. Random Baseline.

List	Mnemonic Phrase
LAPCG	Love after parent combine guerrilla
CHNOPS	Collect happen nice offensive plenty seek
PDNTSPA	Peace design news technical social poverty about
BBROYGBVG WGS	Bus bring red oppose yet grandmother bottom very group white guarantee several

Reading these sequences, we may notice ourselves automatically trying to detect or impose form and meaning on the sequences but should be uncontroversial to say that none of them constitute useful mnemonics. These sequences provide a baseline to appreciate the effects (if any) of our method and the paremeters under investigation.

No POS constraints

The basic capability of the GA and fitness function combination on general words is demonstrated in Table-2 below. The parser and n-grams appear to be enforcing some degree of local syntactic and semantic coherence, which is a noticable improvement over baseline.

Table-2. General Nouns, No POS constraints.

List	Mnemonic Phrase
LAPCG	Look at peace common ground
CHNOPS	Contain high number of people say
PDNTSPA	People develop new to school professor at
BBROYGBVG WGS	Baby boy risk of young girl born violate go with great story

However, if even structurally-poor sentences were to be accepted, there is little memorable in their content. The effect of using famous names instead of general nouns is shown next. After scanning each input list and querying the list of famous names described in 4.1, certain positions in each list were automatically constrained to a number of famous people and associated objects, not exceeding a total of 1/3 of the list length. The results are shown in Table-3.

Table-3. Famous Names, No POS constraints.

List	Mnemonic Phrase
LAPCG	Let Aristotle pay city government
CHNOPS	Cleopatra high number of people Sphinx
PDNTSPA	Plato develop notes to school Pamela Anderson
BBROYGBVG WGS	Below Beyonce risk of young girls below volcano George Washington go see

The grammatical improvement over baseline is also noticable here but content more vivid than that of general words has also been generated for the first and third lists. Unintentionally, the bizarreness of Plato helping to educate Pamela Anderson does seem quite memorable. However, the second and last lists are neither grammatical nor clearly meaningful. The twelve-word list particularly struggles for coherence and seems to demand being split into smaller sentences.

ARPN Journal of Engineering and Applied Sciences

POS constraints

The previous results suggest that the parser and n-grams can enforce some degree of grammatical structure on a sequence but here we pre-constrain the POS sequence for each list using the method described in 3.4.1. Although the structure may be less interesting, it may provide a means to improve meaning. This is shown below for general words.

Table-4. General Nouns, POS constraints.

List	Mnemonic Phrase
LAPCG	Large amounts probably create gas
CHNOPS	Calm husbands never open public schools
PDNTSPA	Perfect days never tell strong public art
BBROYGBVG WGS	Big bright red opinions yearly green broadcast very good way game since

Grammatical constraints have a considerable effect on normalising the sequences into recognisable sentences. Even with general words, the two shorter sequences produce genuine but ambiguous statements. The generated phrase for the chemical elements is actually quite acceptable. The twelve-word sequence is better organised but remains nonsense. Using famous names in a grammatically-sound structure produces the output shown below.

Table-5. Famous Names, POS constraints.

List	Mnemonic Phrase
LAPCG	Liberal Aristotle probably creates Gandhi
CHNOPS	Cleopatra's huge needle only provides Sphinx
PDNTSPA	Plato's dead, never tell sweet Pamela Anderson
BBROYGBVG WGS	Beautiful Beyonce responsibly obeys young girls below vicious George Washington getting service

Again, bizarreness due to the anachronisms of the entities involved is apparent but this is the best output so far. The OSI mnemonic strikes a good balance between oddness and coherence: that someone is dead is probably upsetting so don't tell, but would the Baywatch actress really be concerned about philosopher Plato (no offence to her intellectual sensibilities intended)?

CONCLUSIONS

This paper has described progress on the development of a method for the automatic generation of mnemonic phrases for list information. The experimental results just presented are promising but more work is required before human testing is conducted. There are a large number of parameters in every component of the system that can be varied, both conceptually and in terms of implementation choices. It is too early to say that constraining POS and using famous names is definitively better, but that is the conclusion suggested by these early experiments. It was mentioned earlier that bizarreness was not to be studied in this paper but the experiments produced it anyway. Managing the dissonance between expectations in a controllable way without descending into gibberish will be an interesting challenge. We are also aware that famousness is culturally relative and this suggests the intriguing possibility of more personalised mnemonics. For example, to personalise a mnemonic phrase for the organic elemens for asian learners, words more likely to be memorable might be used: 'Come Home, Neighbour Of President Sukarno'. To personalise the same list for a Briton to remember, an alternative formulation might be more memorable: 'Chelsea Has No Other Players, Sorry!'

REFERENCES

Anslyn E.V. and Dougherty D.A. 2006. *Modern Physical Organic Chemistry*. University Science Books.

Biber B., Conrad S. and Reppen R. 998. *Corpus linguistics: investigating language structure and use*. Cambridge: Cambridge University Press.

Burnage A.G. 1991. *Text Corpora and the British National Corpus*. In Computers and Texts 2, 7-8.

Chomsky. N. 1957. Syntactic Structures. Mouton.

CMU. 2013. http://www.link.cs.cmu.edu/link/.

Davies M. 2011. *The Corpus of Contemporary American English as the First Reliable Monitor Corpus of English*. Literary and Linguistic Computing. 25: 447-65.

Ellis-Ormrod J. 2011. *Educational Psychology, Developing Learners*. Pearson.

Famous. 2013. http://www.whoismorefamous.com/.

Forouzan B.A. 2013. *Data Communications and Networking*. New York, McGraw-Hill.

Gilhooly K.J. and Logie R.H. 1980. Age of acquisition, imagery, concreteness, familiarity and ambiguity measures for 1944 words. *Behaviour Research Methods and Instrumentation.* 12, pp. 395-427.

Glynn S., Koballa T. and Coleman D. 2003. Mnemonic Methods, *Science Teacher.* 70(9): 52-55.

JGAP. 2013. http://jgap.sourceforge.net/.

Jolly W.L. 1991. *Modern Inorganic Chemistry* (2nd ed.). New York, McGraw-Hill.

Klein D. and Manning C.D. 2003. *Accurate Unlexicalized Parsing*. In Proceedings of the 41st Meeting of the Association for Computational Linguistics. pp. 423-430.

Levin J.R., Nordwall M.B. 1992. Mnemonic vocabulary instruction: Additional effectiveness evidence. *Contemporary Educational Psychology.* 17(2): 156-174.

Miller G.A. 1991. *The Science of Words*. W H Freeman and Co.

Mitchell M. 1996. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

N-grams. 2013. http://www.ngrams.info/.

NNDB. 2013. http://www.nndb.com.

Ogden C.K 1937. *Basic English and Grammatical Reform*, Cambridge: The Orthological Institute.

O'Hara R. 2007. *Long-term effects of mnemonic training in community-dwelling older adults*. Journal or Psychiatric Research. 41(7): 585-90.

Russell S. and Norvig P. 2010. *Artificial Intelligence: A Modern Approach*, 3rd Edition. Prentice-Hall.

Sag I. and Wasow T. 1999. *Syntactic theory: A formal introduction*. Stanford, CA: CSLI Publications.

Searleman A. and Herrmann D. 1994. *Memory From A Broader Perspective*, New York: McGraw-Hill.

Seay S.S. 2010. *The Use/Application of Mnemonics As a Pedagogical Tool in Auditing*, Academy of Educational Leadership Journal. 14(2).

Sinclair J. 1991. *Corpus, concordance, collocation*. Oxford University Press.

Sleator D. and Temperly D. 1993. *Parsing English with a Link Grammar*. Third International Workshop on Parsing Technologies.

Stanford. 2013. http://nlp.stanford.edu/software/lex-parser.shtml.

Travers R.M.W. 1982. *Essentials of Learning*. New York, Macmillan.

Voice of America 2013. http://simple.wikipedia.org/wiki/Wikipedia:VOA_Special_English_Word_Book.

Worthen J.B. and Hunt R.R. 2011. *Mnemonology: Mnemonics for the 21st Century*. Essays in Cognitive Psychology, Psychology Press.

Wagner J., Foster J. and Genabith J.V. 2007. *A Comparative Evaluation of Deep and Shallow Approaches to the Automatic Detection of Common Grammatical Errors.* In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 112-121.

Yadav P. 2005. *Electrical Engineering*. New Delhi, Discovery Publishing House.