An Extended System for Labeling Graphical Documents Using Statistical Language Models

Andrew O'Sullivan[1], Laura Keyes[1], and Adam Winstanley[2]

[1] School of Informatics and Engineering, Institute of Technology Blanchardstown, Dublin 15, Ireland
Andrew.O'Sullivan/Laura.Keyes@itb.ie
[2] Department of Computer Science, NUI Maynooth, Maynooth, Co. Kildare, Ireland
Adam.Winstanley@nuim.ie

Abstract. This paper describes a proposed extended system for the recognition and labeling of graphical objects within architectural and engineering documents that integrates Statistical Language Models (SLMs) with shape classifiers. Traditionally used for Natural Language Processing, SLMS have been successful in such fields as Speech Recognition and Information Retrieval. There exist similarities between natural language and technical graphical data that suggest that adapting SLMs for use with graphical data is a worthwhile approach. Statistical Graphical Language Models (SGLMs) are applied to graphical documents based on associations between different classes of shape in a drawing to automate the structuring and labeling of graphical data. The SGLMs are designed to be combined with other classifiers to improve their recognition performance. SGLMs perform best when the graphical domain being examined has an underlying semantic system, that is; graphical objects have not been placed randomly within the data. A system which combines a Shape Classifier with SGLMS is described.

1 Introduction

This paper describes a graphical object recognition framework that applies statistical models to graphical notation based on associations between different classes of object in a drawing to automate the structuring of graphical data. Graphics recognition comprises the recognition and structuring of geometry such as points, lines, text, symbols on graphical documents into meaningful objects for use in graphical information systems for example, Computer Aided Design (CAD), Geographical Information Systems (GIS) and multimedia systems. All of these systems need to capture, store, access and manipulate large volumes of graphical data. For semantic capture of paper/digital data, not only the geometry but also attribute data describing the nature of the objects depicted must be stored, thus representing the graphical data in a high-level object-oriented format for description and semantic analysis. This structuring into composite objects and the addition of labeling attributes is typically a manual, labour intensive, expensive and error-prone process. The automatic structuring and labeling of graphical data is desirable.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 61 – 75, 2006.

This semantic capture and analysis of graphical data is difficult to automate. Graphical object recognition is a sub-field of pattern recognition and includes classification and recognition of graphical data based on shape description of primitive components, structure matching of composite objects and semantic analysis of whole documents. Previous work by authors and colleagues devised and evaluated a graphics recognition system for labeling of objects and components on drawings and plans based on their shape [1]. Shape description has proved successful in distinguishing graphical objects, with classification confidence up to 80% depending on the domain, however, no one shape method provides an optimal solution to the problem. Automation of the structuring and recognition of objects through statistical modeling for efficient and complete input into graphical information systems can form a solution to this complex problem. That is, treating the graphical document as a language, statistical language modeling is applied through a statistical graphical language model framework.

Statistical Language Models (SLMs) are successful methods used in Natural Language Processing (NLP) for recognising textual data. SLMs estimate the probability distributions of letters, words, sentences and whole documents within text data. They have been used for, among other tasks, Speech Recognition [2] and Information Retrieval [3]. This work investigates the use and adaptation of SLM techniques that is, Statistical Graphical Language Models (SGLMs) to aid in the semantic analysis of graphical data on graphical documents. The proposed framework will apply statistical models to graphical languages (CAD data) based on the associations between different classes of shape in a drawing to automate the structuring of graphical data and to determine if SLMs have applicibility to improve the classification of graphical objects as they do for NLP applications. A SGLM module to extend the system for labeling and semantic analysis of graphical documents to improve performance is applied.

In this paper, SGLMs for graphics recognition is presented. Section 2 describes SLMs as a method used in natural langauge processing and their application to graphical data. It outlines the similiarities between natural language and the language characterised by graphical data that support the application of SLM to graphical notation and shows how N-gram models, a widely used SLM technique, can be used to build SGLMs for the recognition of unknown objects within CAD drawings for engineering plans. Section 3 depicts the graphical recognition system used and the application of the SGLM module to extend the system for labeling and semantic analysis of graphical documents. Section 4 describes the background to this work, the experimental work carried out and discusses the results. Section 5 concludes and outlines future work.

2 SLMs and Graphical Object Recognition

Statistical Language Models (SLMs) are estimates of probability distributions, usually over natural language phenomena such as sequences of letters, words, sentences or whole documents. First used by Andrei A. Markov at the beginning of the 20th century to model letter sequences in Russian literature [4], they were then developed as a general statistical tool, primarily for NLP. Automatic Speech Recognition is arguably the area that has benefited the most from SLMs [2] but they have also been used in many

other fields including machine translation, optical character recognition, handwriting recognition, information retrieval and augmentative communication systems [5].

There are different types of SLMs that can be used. These include *Decision Tree* models [6], which assign probabilities to each of a number of choices based on the context of decisions. Some SLM techniques are derived from grammars commonly used by linguists. For example Sjilman et al. [7] use a declarative grammar to generate a language model in order to recognise hand-sketched digital ink. Other methods include *Exponential* models and *Adaptive* models. Rosenfeld [8] suggests that some other SLM techniques such as *Dependency* models, *Dimensionality* reduction and *Whole Sentence* models show significant promise. However this research will focus on the most powerful of these models, *N-grams* and their variants.

2.1 N-gram Models for Predicting Unknown Words in NLP

N-gram models are the most widely used SLM technique. In NLP N-grams are used to predict words based on their N-1 preceding words. The most commonly used N-grams are the bigram model, where N=2 and the trigram model, where N=3. That is, a bigram model uses the previous word to predict the current word and a trigram model uses the two previous words. These probabilities are estimated by using the relative frequencies of words and their co-occurrences within a training corpus of natural language examples.

For bigram models, the corpus of data is analysed for the relative frequencies of pairs of words that occur together. For instance if the last sentence was analysed the following pairs would be recorded: “For bigram”, “bigram models”, “models the” and so on. The same applies for trigram models, except the corpus is analysed for triples, not pairs, of words that occur together. Bigram tables and trigram tables store these frequencies, which are then used to predict unknown words. These probabilities can be estimated using the equations (1) and (2), respectively.

$$P(w_i \mid w_{i-1}) = C(w_{i-1}\, w_i) / C(w_{i-1}) \tag{1}$$

$$P(w_i \mid w_{i-1},\, w_{i-2}) = C(w_{i-2}\, w_{i-1}\, w_i) / C(w_{i-2}\, w_{i-1}) \tag{2}$$

where C represents the frequency of words occurring together. The right hand sides of equations (1) and (2) are computed from the bigram and trigram tables, correspondingly. The w_i that results in the highest frequency and hence the highest probability is judged to be the next word in the sentence. The corpora required for this process are usually extremely large and contain a wide range of examples of natural language. For example the Brown Corpus [4] contains one million words taken from fifteen different sources such as legal text, scientific text and press reportage. It should be noted however that corpora can be constructed to just include a particular subset of language, if so required for a particular task.

2.2 SLMs for Labeling Graphical Documents

SLMs have previously been used almost exclusively for NLP. There are sufficient similarities between natural language and graphical notations that suggest that adapting SLMs to become SGLMs is a worthwhile approach [9].

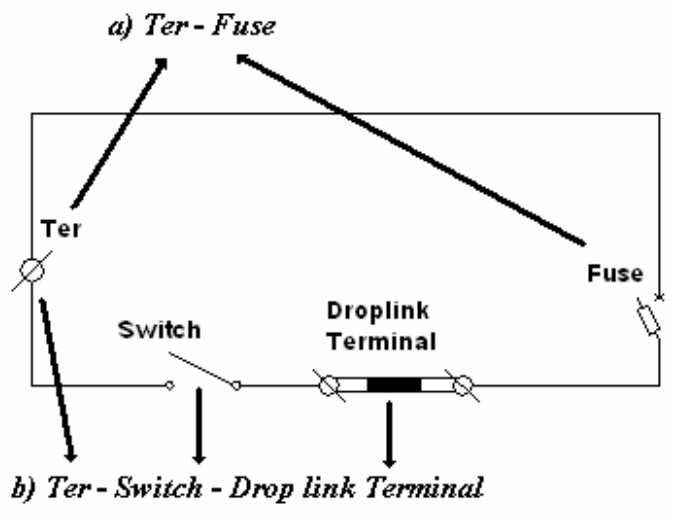

Fig. 1. Sample electrical circuit and phrases constructed

Recent work applied SLMs to the automatic structuring of topographic data [10] for Geographical Information Systems (GIS). In their work Winstanley and Salaik characterise the similarities that can be drawn between topographic data and natural language. Both consist of discrete objects (words, graphical objects) and these objects:

- have a physical form (for example spelling, object shape);
- have a semantic component (meaning, graphical object label);
- are classified according to function (part of speech, object class) and
- are also formed into larger components (sentences/paragraphs, diagrams/ documents).

A similar analogy can be used for natural language and graphical data found on architectural or engineering plans. By considering the graphical data as a language with its own syntax and vocabulary the analogy becomes:

- Word – particular object
- Spelling – configuration of graphic components of object (shape)
- Part-of-speech – type of object (relay, resistor)
- Phrase – connected sequence of objects

Using this framework N-gram models can be constructed that build phrases representing graphical data on drawings and plans. Figure 1 shows a sample circuit and phrases that can be constructed for a graphical language.

2.3 SGLMs for Labeling Graphical Documents

As in NLP, a corpus of training data is needed for SGLM. This training data must contain examples of graphical objects in their contextual use that is, actual real world documents. N-gram tables must be built which contain the relative frequencies of

co-occurrences of the graphical objects. This requires the counting of occurrences of phrases of objects within the corpus. It is here that one of the major differences between natural language and graphical notations is noted. Natural language is a one-dimensional sequence of symbols, whereas graphics are inherently multi-dimensional. This difference is significant in relation to N-grams as the one-dimensionality of natural language makes the choice of which words to use for phrase construction and counting an easy one that is, the preceding words of the unknown word. With graphical notation however, there can be numerous other objects neighbouring the unknown object. This makes the choice of which of these neighbouring objects to use to construct object phrases a harder one. One approach to dealing with this is to use adjacency relationships between objects on a document.

2.4 Object Adjacencies

In SGLMs, neighbouring objects are used to form object phrases. How the term *neighbouring* is defined will govern how the object construction process works. Object *adjacencies* are used for this purpose, with the adjacencies defining how objects relate to each other. Once an adjacency is defined for a particular domain or diagram all the objects within that data that are adjacent to one another can be used to form object phrases, for storage in the *N*-gram tables. Defining the object adjacency rules that will govern how the object phrases are constructed is an important decision in designing SGLMs. There are several ways to define *adjacent* in this context. Experimental work undertaken so far has used a corpus of graphical documents consisting of electrical circuits. Objects are defined as being *adjacent* to one another if they are connected by a wire. For example in Figure 1 two examples of phrases of objects that can be constructed using this adjacency definition are shown. Part a) shows the constructed bigram phrase "*Ter – Fuse"* and part b) shows a trigram phrase "*Ter – Switch – Droplink Terminal"*.

This method of defining adjacency does not take into account higher-level information about electrical circuit diagrams. For example objects within circuits can occur in series or parallel with each other. Objects in series relate differently to other objects than if they were in parallel. This suggests that the object *adjacencies* should

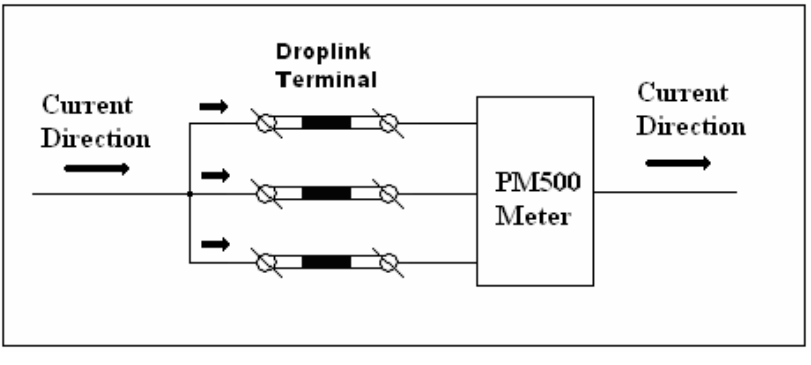

Fig. 2. Electrical Circuit example with current direction indicated by black arrows

attempt to model these differences, perhaps by having more than one type of adjacency. This however would introduce more complexity into the process and for the experimentation conducted so far objects in parallel were not treated differently to objects in series. Possible solutions to this problem involve determining different ways of defining object neighbours and counting phrases. Options include the use of direction in the adjacency definition. For example object phrases could only be formed in the direction of the current. So in Figure 2 below, the Droplink Terminal Objects would only form phrases with the PM500 Meter object and not each other. This is ongoing work being investigated by authors in current experiments.

2.5 N-gram Models for Predicting Unknown Graphical Objects

All of the phrases extracted from the corpus are used to build bigram and trigram frequency tables. The frequencies of phrases are known but the relative frequencies must be obtained as they estimate the probabilities. The relative frequencies of N-gram phrases are computed by dividing the frequency of a phrase by the total frequency of that phrase. Relative frequencies for bigram and trigram phrases are computed using equations (1) and (2) in section 2.1. The resulting bigram and trigram tables are used to predict unknown objects.

One problem associated with N-grams is the data sparseness problem. This means that there are some events within the N-gram tables that have a probability of zero. This is because those events did not occur in the training data so they have a frequency and hence probability of zero. These events therefore will not be considered in any future prediction process, even though the events may actually occur in the future. The data set used in this work is limited in terms of size so such zero-probabilities were expected. However, even with extremely large datasets, zero-probabilities occur. A solution to this problem is *Smoothing*. *Smoothing* attempts to give probability values to events with zero probability. There are several *Smoothing* techniques available but here *Add-One Smoothing* is used. This is a simple technique where the value '1' is added to all the entries in the bigram and trigram frequency tables. So any event, which previously had a zero frequency, will now have a frequency and a probability.

3 Graphics Recognition System with SGLM

This work investigates the use and adaptation of SLM techniques i.e. Statistical Graphical Language Model (SGLMs) to aid in the semantic analysis for structuring and labeling graphical data on technical documents for the purposes of recognition, indexing and retrieval. An earlier system has been developed for the recognition and labeling of graphical objects where the underlying classifier is based on shape recognition [1]. Shape methods are applied to object boundaries extracted from drawings represented as vector descriptions.

3.1 Shape Classifier

To assess the capability of the SGLM to improve the performance of other classifiers, a classifier was implemented which is based on simple set of shape descriptors. The

shape classifier, implemented in Matlab, uses the following six descriptors to classify the graphical entities:

- Bounding Box width to height ratio. The bounding box is the smallest rectangle to enclose the symbol.
- Minor Axis Length to Major Axis Length ratio. The length of the minor and major axis' of the ellipse that has the same second-moments as the region.
- Eccentricity. The ratio of the distance between the foci of the ellipse and its major axis length.
- Euler Number. The number of objects in the symbol minus the number of holes in those objects.
- Solidity. The proportion of the pixels in the convex hull that are also in the symbol.
- Extent. The proportion of the pixels in the bounding box that are also in the symbol.

The output obtained by the description methods provides a measurement of shape that characterises the object type. These shape descriptors provides a list of candidate classes of each object. Extending this system with SGLM is envisaged as a possible means of improving the performance of the overall graphical object recognition system. The SGLM model is combined with the score produced by shape classifier to improve the likelihood that the classification is correct or re-classify incorrect or misclassified features. Figure 3 shows the configuration of the recognition system and the role of SGLM within this system.

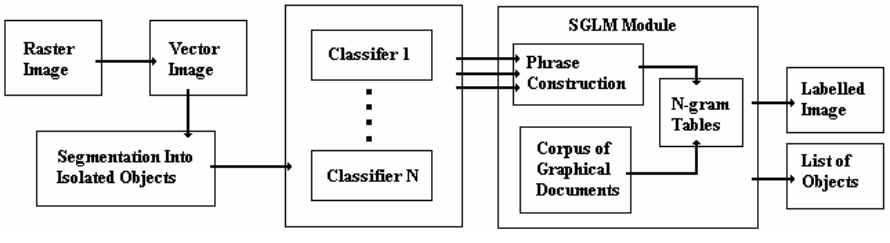

Fig. 3. Extended Recognition System

3.2 Combining N-grams with Shape Classifier

It is suggested by this research that the main benefit of adapting N-grams to work for graphical notations is in improving the performance of other classification technique. It is proposed to use the developed N-grams to improve the performance of a shape classifier. In a document for each unknown object the shape classifier produces a candidate list of possible identities. These possible identities and the identities of the object's neighbours are then used to construct object phrases. The N-gram tables are then consulted to find the most probable of these phrases and hence find the most probable identity for the unknown object. For example, Figure 4 shows the same circuit as in Figure 1, except that the switch's identity is unknown. There are two possible trigram phrases involving the unknown object:

- "*Fuse – Ter – Unknown Object*"
- "*Fuse – Droplink Terminal – Unknown Object*"

Table 1. Sample shape classification of unknown object

Classification	Probability
Isolator	0.4536
Switch	0.3241
Ter	0.1532
ELU	0.0072

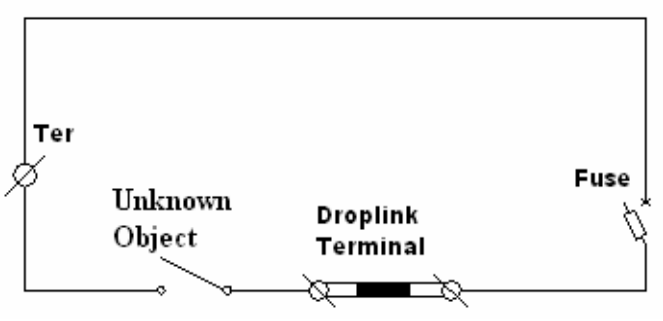

Fig. 4. Sample of circuit with unknown object

Table 1 shows sample results of shape classification for the unknown object. Combining these with the two trigram phrases taken from Figure 4 gives eight candidate phrases:

- *"Fuse – Droplink Terminal – Isolator"*
- *"Fuse – Droplink Terminal – Switch"*
- *"Fuse – Droplink Terminal – Ter"*
- *"Fuse – Droplink Terminal – ELU"*
- "*Fuse – Ter – Isolator*"
- *"Fuse – Ter – Switch"*
- *"Fuse – Ter – Ter"*
- *"Fuse – Ter – ELU"*

The trigram table can now be checked to see which of the eight phrases is the most frequent and hence which identity to assign the unknown object.

The combination of shape classification and N-grams described, form a major part of this projects work. Another major part will develop Part-of-Speech (POS) tagging for use with the graphical notation. POS tagging is a technique that is used in NLP to assign tags to words. Examples of these tags are noun, verb adjective and pronoun. Tags can be assigned to graphical objects by using the equation:

$$P(\text{object shape} \mid \text{tag}) * P(\text{tag} \mid \text{neighbouring k tags}_i) \quad (3)$$

This is the probability of an object belonging to a particular class combined with the likelihood that that class would have the observed neighbouring class (of neighbours up to *k* deep).

Part of this research will be to ascertain the best way to define tags for graphical objects. For example, with the electrical data, tags could be based on hierarchical classes e.g. an object is identified as a Resistor and its tags are the various types of Resistor such as 10 Ohm, 20 Ohm etc. Another approach could be to define tags based on the object's function within the circuit e.g. an object could be a Meter, a Relay or a User.

4 Experimental Work

Experimental work to determine the applicability of SLMs to graphical data was carried out in two phases. Firstly SLMs were applied on their own to the data. This initial step was used to establish the feasibility of applying SLMs in the form of SGLMs to deal with a graphical language. Secondly the SGLMs were combined with a set of simple shape classifiers and the results evaluated. The following sections outline each experiment and discuss the results obtained. Figure 5 shows sample vocabulary of graphical language used in this work. The amount of data available for use in this work was limited at the time of these experiments; however, this work and corpus of data constructed are used to determine the viability of this novel approach to graphical object recognition.

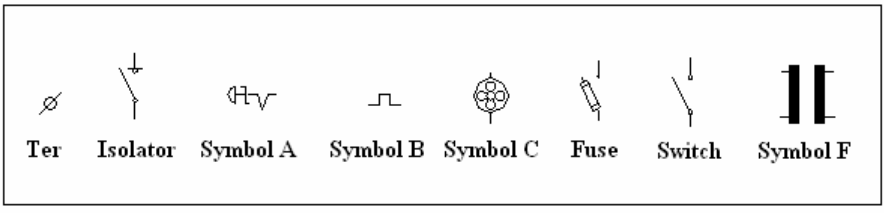

Fig. 5. Sample vocabulary used in experiment

4.1 Effectiveness of SGLMs on CAD Data

This works forms part of a project to develop an online Operation and Maintenance information system. The O&M System allows a user to select an example object (simple or composite) and the software finds similar objects in the same or other drawings. The tool generates data structures that can be used to build multimedia linkages between objects, drawings and related information. The information is accessed through a standard web browser interface including navigation through hot-links and keyword search facilities. CAD drawings showing the location of utilities and services also act as browser navigational maps. The system can be implemented for all sizes of installations but comes particularly suited for the infrastructure management of large industrial or service sites. Current use relevant to this paper is electrical data for business park sewage pumping station.

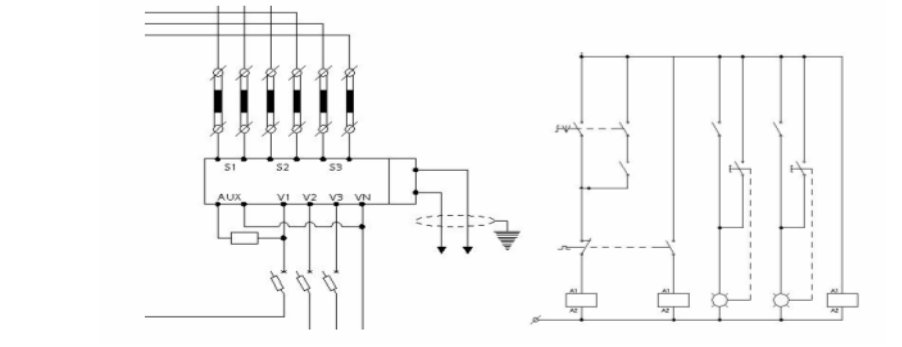

Fig. 6. Samples of the electrical circuits used in the work

Figure 6 shows examples of electrical data used in this experiment. The graphical data used for this work consists of electrical circuits. In total 18 electrical diagrams were used. The diagrams contain 738 graphical objects (excluding wire connection lines) and there are 24 different objects types. A bigram model and a trigram model were implemented on the graphical notation. For the bigram model phrases of pairs of objects which occurred together within the data were counted. Likewise for the trigram model triples of co-occurring objects were counted. The bigram phrases were stored in a 2-dimensional array, where the index *(i, j)* corresponds to the number of times the Object_i occurred with the Object_j. The trigram phrases were stored in a similar 3-dimensional table.

The N-gram tables were tested on two unseen electrical diagrams. The first diagram tested contained 39 objects and 8 object types. The second diagram contained 30 objects and 6 object types. Each object was treated as an unknown object and its adjacent objects were used to construct bigram and trigram phrases. The probabilities of these phrases were then combined into one final prediction for each object by three different voting combination methods: Majority Vote, Sum Rule and Maximum [11]. Table 2 shows the performance results of the bigram and trigram models in terms of the percentage of objects they classified correctly. Table 3 shows a more detailed breakdown of the trigram model's performance with the first diagram.

4.1.1 SGLMs Results Discussion

These experiments were used to determine the applicability of applying SLMs to graphical CAD data. N-grams are not primarily designed to work on their own so the low percentage rates of objects correctly predicted are not unusual. The small size of the test data is also an obvious factor in the results. As the project continues and the test data is enlarged with more object types and contextual use examples added, the performance of the N-grams should improve.

Table 2. Bigram and trigram performance results

N-gram:	**Bigram**			**Trigram**		
Combination Method:	**Majority**	**Sum**	**Max**	**Majority**	**Sum**	**Max**
Drawing 1:	33%	30%	13%	44%	49%	44%
Drawing 2:	30%	30%	17%	37%	37%	17%

Table 3. Detailed trigram performance results for Drawing 1

Object Type	Total Number of Objects	Amount Predicted Correctly			Percentage Predicted Correctly		
		Majority	Sum	Max	Majority	Sum	Max
Switch	18	11	11	11	61	61	61
Symbol A	2	0	0	0	0	0	0
Symbol B	3	0	0	0	0	0	0
Symbol D	1	0	0	0	0	0	0
Symbol E	3	0	0	0	0	0	0
Ter	6	0	2	0	0	33	0
ELU	5	5	5	5	100	100	100
HOA	1	1	1	1	100	100	100

When the N-grams were used on their own on the electrical diagrams they displayed typical behavior. N-grams are highly sensitive to their test data, with objects or events with high frequencies within the test data predicted with large frequency during classification processes. For example many objects were misclassified as *Switch* during the testing as *Switch* is one of the most frequent objects within the data. Likewise, as Table 3 shows, none of the entities of object type *Symbol A, Symbol B, Symbol D or Symbol E* were correctly predicted. This is due to their low frequency within the test data. The objects of type *ELU* however, which have high frequency values, were 100% correctly predicted.

The trigrams performed better than the bigrams, again this was expected as bigrams use less information than trigrams who use two neighbouring objects to form a phrase. If N was increased to four, to form a Quadgram table, the performance could be improved further. The complexity of the process however would be increased significantly.

4.2 Shape Classifier Combined with SGLMs

The SGLMs combined with the shape description approach were tested on two electrical diagrams, consisting of 43 electrical symbols and 14 symbol types. For each symbol the shape classifier produces a ranked list of possible symbol types. The candidates are ranked based on the distance between the unknown symbol's descriptor values and the ground truth values of the symbol types. In this classification if the top 2 ranked scores are within 5% the classification is deemed to be uncertain. And the SGLMs employed. The unknown symbol's neighbours are used to create symbol

phrases. The symbol, which in combination with the symbol phrases has the highest frequency value within the training data, is judged to be the identity of the unknown symbol.

Table 4. Recognition perfromance results

		Recognition Performance: % recognised correctly			
Object Type	**Amount in Test Data**	**Shape**	**Shape + Trigram**	**Shape + Bigram**	**Shape + (Trigram When Shape Uncertain)**
Ter	8	25	62.5	0	62.5
Mi-crologic	3	10	66.67	0	100
ELU	8	87.5	100	0	87.5
Symbol A	2	100	0	0	100
Symbol B	1	0	0	0	0
Symbol C	1	100	0	0	100
Symbol E	1	100	0	0	100
Symbol F	3	100	0	0	100
Switch	9	66.67	66.67	66.67	88.89
Fuse	2	100	100	0	100
Droplink	1	100	100	100	100
PM500	1	100	100	0	100
Isolator	1	100	100	100	100
ASP	1	100	0	0	100
HOA	1	100	100	0	100
Total	**43**	**74.4**	**62.79**	**18.6**	**86**

4.2.1 Combined Approaches Results Discussion

The shape classifier recognised 32 of the 43 symbols correctly, a rate of 74.4%. When the Trigram SGLM module is used in combination with the shape classifier on every symbol, 27 symbols are recognised correctly, which at a rate of 62.79%, is a decrease of 11.61%. There is a decrease in recognition performance because the SGLM module typically fails to recognise symbols that have low frequency within the training data. For example, Symbols A, B, C E and F occur relatively infrequently within the training data and as seen in Table 4 the SGLM failed to recognise them within the test data. When the bigram SGLM is used the recognition rate falls severely to 18.6%. This is to be expected as bigram models typically perform worse than trigram models as they make use of less information. In this case the information in question is the identities of the neighbouring symbols. The low bigram recognition rate can be

viewed as proof that by using more of the neighbouring symbols the performance of the SGLM improves.

When the trigram SGLM module is used in combination with the shape classifier only when the shape classifier is uncertain about classification, 37 of the symbols are recognised. This is a recognition rate of 86%, which is a 11.6% increase in recognition from the original rate of 74.4%. By using the SGLM only when the shape classifier is uncertain, symbols that the SGLM might fail to recognise have the chance to be recognised by the shape classifier. Likewise, when the shape classifier is uncertain, the symbols in question have a chance to be recognised by the SGLM. This method of using both recognition techniques has resulted in an increase in recognition performance, which shows promise for the use of SGLM.

It should be noted that in this test, it is assumed that when the SGLM module is used to classify a symbol, the identities of the neighbouring symbols are known. This of course might not be the case so an option is to use the shape classifier to temporarily classify any unidentified neighbouring objects and use these temporary identities for use with the SGLM for the current symbol. Further tests will assess the performance of this approach.

Another factor of interest is the number of neighbouring symbols to use. At present the bigram and trigram models have been used, which use one and two neighbouring symbols respectively. An increase in this number could result in improved results, as more information is used. A Quadgram for example would form three-symbol phrases from the neighbouring symbols. This increase however would result in an increase in computational expense. One problem with forming symbol phrases within the electrical domain that is the focus of these tests is the number of wire connections within the electrical circuits can result in a large number of phrases being created. An increase in the number of symbols used could result in an even larger number of phrases created, which increases the computational expense.

5 Conclusion and Future Work

This paper has proposed the adaptation of Statistical Language Models for recognition and labeling of graphical objects within architectural and engineering documents. Previously used for Natural Language Processing there exists similarities between natural language and technical graphical data that suggest that Statistical Graphical Language Models is a worthwhile approach. Digitised CAD drawings for electrical data are processed to extract their component objects. SLM are applied and N-gram phrases are constructed. Initial experiments apply SGLM without the combination of other classifiers to determine their applicability and effectiveness at classifying graphical objects. Results show classification rates of less than 50% for bigram model and 61% and 100% for certain instances of graphical objects using trigram model. The size of data used in the experiment is a factor in results. However, it is envisaged that with bigger amounts of data for training and testing and increased frequencies of graphical objects in data, the performance of SGLM will improve.

The SGLMs are designed to be combined with other classifiers to extend previous recognition system. Using this approach SGLMs are applied based on associations between different classes of ‘shape’ in a drawing to automate the structuring and

labeling of graphical data. Digitised CAD drawings are processed to extract their component objects from which shape descriptions are built. These feed into several description and matching algorithms, each of which produces one or more candidate categories to which each object may belong. An overall consensus decision gives a ranked list of candidate types. The SGLM module can then be used to improve the performance of the recognisers.

Combination of scores can take the form of voting methods such as majority vote or borda count. An extension of N-grams used in NLP is to count the part-of-speech of the word (noun, verb and so on) rather than the word itself. This n-gram part-of-speech tagging model can be used with shape for graphical data, where the tag is some descriptive classification of the graphical object. It s envisaged that tagging will provide a effective means of combining SGLM module with the existing graphical recognition system.

The experiments conducted so far to evaluate SGLMs have been conducted on a limited dataset. Training corpora used in Natural Language Processing however, can contain millions of words. The next stage in evaluating SGLMs is to undertake a large-scale experiment, with a significantly larger number of graphical diagrams and objects used. The authors are currently undertaking this experiment with electrical circuit diagrams. There is a vastly increased vocabulary of graphical objects being considered and as such the number of circuit diagrams needed is also immensely increased. Whereas the previous experiments involved 18 diagrams, the present research involves thousands.

Different approaches to the adoption and application of SGLM will be carried out. Other possibilities include different ways of defining the *adjacency* of objects, Different vote combination methods such as Borda Count, Minimum and Median will be computed to find the optimal method. Part-of-Speech tagging as a way of combining modules will be exhaustively tested. A final SGLM module can be used to extend and improve the performance of system for the labeling and semantic modeling of graphical documents.

References

1. Keyes, L., Winstanley, A., "Shape Description for Automatically Structuring Graphical Data", in Josep Llados, Y.B. Kwon (eds), Graphics Recognition – Recent Advances and Perspectives, LNCS 3088, 353-262, Springer-Verlag, 2004
2. Jelinek, F., Statistical Methods for Speech Recognition. MIT Press 1997
3. Ponte, J.M., Croft, W.B., "A Language Modeling Approach to Information Retrieval", Proceedings of SIGIR'98, 1998, 276-281
4. Manning, C.D., and Schutz, H., Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, 2001.
5. Jurafsky, D. and .Martin, J.H., Speech and Language Processing, Prentice-Hall, 2000.
6. Bahl, L R., Brown, P. F., Peter V. de Souza and R. L. Mercer., "A Tree-based Statistical Language Model for Natural Language Speech Recognition." IEEE Transactions on Acoustics, Speech and Signal Processing, 37:1001-1008, July 1989.
7. Shilman, M., Pasula, H., Russell, S. and Newton, R., "Statistical Visual Language Models for Ink Parsing." AAAI Spring 2002 Symposium on Sketch Understanding, 2002.

8. Rosenfeld, R., “Two Decades of Statistical Language Modeling: Where Do We Go From Here?”, Proceedings of the IEEE, 88 (8), pp 1270-1278, 2000.
9. Andrews, J.H., Maps and Language, A Metaphor Extended, *Cartographic Journal*, 27, 1–19, 1990.
10. Winstanley, A., B. Salaik, L. Keyes: “Statistical Language Models For Topographic Data Recognition”, IEEE International Geoscience and Remote Sensing Symposium (IGARSS’03), July 2003.
11. J. Kittler, M. Hatef., R.P.W. Duin and J. Matas, “On Combing Classifiers” IEEE Transaction on Pattern Analysis and Machine Intelligence, 20 (3), 226-239, 1998.