Information Fusion in Visual-Task Inference

Amin Haji-Abolhassani
Centre for Intelligent Machines
McGill University
Montreal, Quebec H3A 2A7, Canada
Email: amin@cim.mcgill.ca

James J. Clark
Centre for Intelligent Machines
McGill University
Montreal, Quebec H3A 2A7, Canada
Email: clark@cim.mcgill.ca

***Abstract*—Eye movement is a rich modality that can provide us with a window into a person's mind. In a typical human-human interaction, we can get information about the behavioral state of the others by examining their eye movements. For instance, when a poker player looks into the eyes of his opponent, he looks for any indication of bluffing by verifying the dynamics of the eye movements. However, the information extracted from the eyes is not the only source of information we get in a human-human interaction and other modalities, such as speech or gesture, help us infer the behavioral state of the others. Most of the time this fusion of information refines our decisions and helps us better infer people's cognitive and behavioral activity based on their actions. In this paper, we develop a probabilistic framework to fuse different sources of information to infer the ongoing task in a visual search activity given the viewer's eye movement data. We propose to use a dynamic programming method called *token passing* in an eye-typing application to reveal what the subject is typing during a search process by observing his direction of gaze during the execution of the task. Token passing is a computationally simple technique that allows us to fuse higher order constraints in the inference process and build models dynamically so we can have unlimited number of hypotheses. In the experiments we examine the effect of higher order information, in the form of a lexicon dictionary, on the task recognition accuracy.**

***Keywords*-attention; visual search; cognitive modeling; task inference; information fusion; eye movement;**

I. Introduction

The link between eye movements and visual task has enjoyed burgeoning attention in psychophysical and cognitive sciences. Particularly the effect of visual task on parameters of eye movements have been investigated for a long time in the literature. In two seminal studies, Yarbus [1967] and Buswell [1935] showed that visual task has a great influence on specific parameters of eye trajectory. Figure 1 shows Yarbus's observation that implies fixation locations are not randomly distributed in a scene but instead tend to cluster on some regions at the expense of others. In this figure we see how visual task can modulate the conspicuity of different regions and as a result change the pattern of eye movements. Based on this experiment, the effect of task on the pattern an parameters of eye movement is called the *forward Yarbus process*.

The forward Yarbus process is also studied in a work by Clark and O'Regan [1998], who examined the dynamics of eye movements when reading a text. They showed that when reading a text the centre of gaze (COG) lands on the locations that minimize the ambiguity of the word arising from the incomplete recognition of the letters. In another study Castelhano et al. [2009] showed that different patterns of eye movements emerge from tasks of memorizing a scene and searching for an object in it.

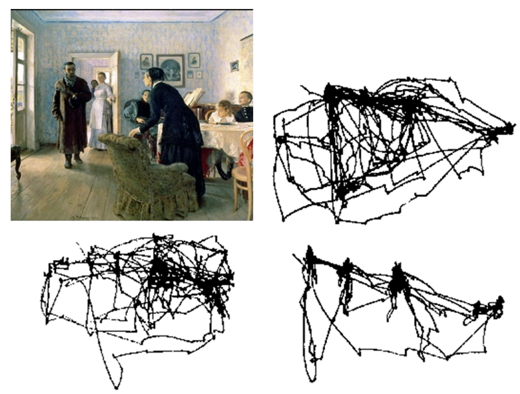


Figure 1: Eye trajectories recorded by Yarbus while a viewer carried out different visual tasks. Upper right - no specific task, lower left - estimate the wealth of the family, lower right - give the ages of the people in the painting [Yarbus, 1967].

In a forward Yarbus process the visual task is given as an input and the output is task-dependent scanpaths of eye movements. The other way of looking at the interaction between eye movements and visual task is to study the reverse path from the scanpaths to the visual task. The inference of task from eye movement data is called an *inverse Yarbus process* and has recently gained a growing interest in psychophysical studies of human attention.

Visual search is one of the main ingredients of human vision that plays an important role in our everyday life. Recently in [Haji-Abolhassani and Clark, 2011a,b] we proposed a model based on the theory of Hidden Markov Models (HMMs) to infer what the viewer is looking for in a task of searching for pop-out objects in digitally created stimuli. In [Haji-Abolhassani and Clark, 2012a,b] we extended our model to infer the word that the viewers typed using their eye movements (*eye-typing*) in a soft keyboard application. In both scenarios the viewer executes visual search and the model calculates a probability distribution on different possible tasks given the eye data, and makes an inference about what objects are being sought using *maximum likelihood* (ML). In real life, however, we incorporate a-priori sources of information in the ML estimator and make inferences based on *maximum a-posteriori* (MAP) estimation. For instance, when looking for an orange in a basket of fruits, the prior knowledge about the color of oranges helps us skip the objects with different colors and narrow down our search to the orange areas.

In this paper we extend our HMM-based model presented in [Haji-Abolhassani and Clark, 2012a,b]; which we will be referring to as tri-state HMM (TSHMM) in the rest of the text; to incorporate a-priori information to infer the visual task in the eye-typing application. In order to infer the ongoing task, we propose to use the TSHMM model within a simple conceptual model of eye movement recognition based on a technique called *token passing* that incorporates the TSHMMs in a transition network structure. In the new structure, the higher order constraints are applied along transitions from a TSHMM unit to another. Moreover, since in token passing method the models are generated dynamically during the test phase, we can have an unlimited number of hypotheses in our experiments.

In the following sections we will first revisit the TSHMM model used for task inference in the eye-typing application. Then we show how we can equip the model with high level constraints. In the experiments we show how using a-priori information in the form of a lexicon dictionary improves the recognition rate.

II. Task inference using Hidden Markov Models

The application we designed for task inference in visual search is an eye-typing application, where subjects can type a word by directing their gaze on its comprising characters. Figure 2 shows the schematic of the on-screen keyboard used in the experiments. We removed the letter "Z" from the keyboard to obtain a square layout to reduce directional bias. In order to impose visual search, we randomized the location of characters to eliminate any memory effect.

Although visual attention and direction of gaze are sometimes assumed to be the same, in oculomotor studies of human vision it is shown that the focus of attention (FOA) can be well away from the center of gaze (COG) [Fischer and Weber, 1993]. Based on the alignment of the COG to the FOA we have two types of visual attention; that are covert and overt attention. In overt visual attention the FOA is aligned to the COG and in the covert visual attention the FOA is away from COG.

Figure 2: The schematic of the on-screen keyboard used in the eye-typing experiments. We removed the letter "Z" in order to have a square layout to reduce directional bias. Also the location of each character is randomized in each layout so that the user has to search for the characters.

Perhaps the first scientist to provide an experimental demonstration of covert attention is known to be Helmholtz [1896]. In his experiment, Helmholtz briefly illuminated inside a box by lighting a spark and looked at it through two pinholes. Before the flash he attended to a particular region of his visual field without moving his eyes in that direction. He showed that only the objects in the attended area could be recognized implying attention can be away from the eye movements.

Apart from the intrinsic difference between the FOA and the COG in covert shift of attention, the focus of overt attention can also be different from the COG reported by the eye-tracker due to the noise of the recording instrument. Moreover, overshooting or undershooting of the targets can cause a mismatch between the COG and FOA, regardless of the attention type, which urges us to allow for discrepancy between these two phenomena in the attention models.

Hidden Markov Models (HMMs) are a class of generative methods that are used to classify sequential observations. A typical HMM is composed of a number of *states* that are hidden from the observer. The transitions between the states are governed by a *transition probability matrix*, A, that gives us the chances of transitions from a state to the connecting states. At each time-step, an observation is generated according to an *observation probability density function* that is assigned to the current state. The observation pdf is characterized by a set of parameters B that defines the properties of the pdf. At the beginning of each sequence, the HMM selects a starting state according to a *initial state distribution*, Π, and carries on by chooses the next states at each time-step according to A. Figure 3 shows a sample tri-state HMM with its corresponding observation pdf, transition probabilities and initial state distribution.

HMMs have been extensively used in the field of speech recognition [Rabiner, 1990], optical character recognition [Hu et al., 1996] and anomaly detection in video surveillance [Nair and Clark, 2002] before. There are usually three different problems that are addressed in the literature

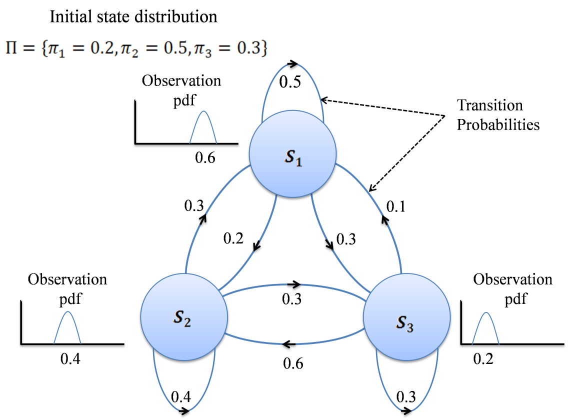


Figure 3: A sample first-order, discrete-time, continuous HMM. An HMM is defined by its number of states, transition probabilities, observation pdfs and initial state distribution in a tuple $\lambda(A, B, \Pi)$.

related to the HMMs; that are training, decoding and evaluation. The training is done by an algorithm called *Baum-Welch*, whereby we train the parameters of HMMs using the training data. In decoding, the best sequence of states is revealed by using a method called *Viterbi*. Finally, in the evaluation, we use a method known as *forward algorithm* to find the likelihood of a an observation given the parameters of an HMM.

In the TSHMM model we used HMMs to model the cognitive process of human brain that controls the COG and FOA. In the model we represented the FOA by the hidden states of an HMM and the observations of the HMM were equivalent to the COG. The only information we observe from a human eye is the COG and the FOA is hidden from us. This is inline with the structure of HMMs, where we only see the observations and the states are hidden to the observer.

Looking for a character among other characters is a visual search task that requires a combination of features to be used to locate the target [Treisman and Gelade, 1980]. This characteristic calls for an attentive, mainly serial, limited capacity attentional deployment over a limited portion of the visual field which usually entails several fixations on distractors (non-targets) before locating a target. Moreover, during the experiments we observed a pattern in these off-target fixations that implies the FOA doesn't randomly scan the characters to seek a target, but instead tend to verify the similar characters more often than dis-similar ones. This effect is studied before in perceptual measurement of image similarity in [Keren and Baggen, 1981, Gilmore et al., 1979]. Figure 4a shows the result of an experiment in [Gilmore et al., 1979] that categorizes the characters according to their similarity. In this figure a hierarchical clustering is used to classify characters according to their similarity (i.e., the lower the connecting line between clusters, the higher the similarity between the clusters). Figure 4b shows that a similar pattern appears in our experiments.

Based on these facts we designed the TSHMM model so that is allows for attentional deployment both on target and non-target objects. Furthermore, we divided the off-target fixations according to their similarity to the target. In figure 5a we see the proposed attention model for the task of looking for a character. For each character we train an HMM that has three states that represent deployment of attention on non-target, similar-to-target (*S-state*); non-target, dis-similar-to-target (*D-state*); and target characters (*T-state*). As we showed, this structure elicits more information from off-target fixations which increases the accuracy of task inference.

The observation pdfs generate COGs according to the attention state and are defined by GMMs with equal weights in the D-state and S-state, and a single Gaussian in the T-state. The GMM of the S-state has a mean vector that points to the top two similar characters according to the fixation frequency histogram (similar to figure 4b) that is obtained in the training phase. The GMM of the D-state is simply the negation of the S-state and T-state's observation pdfs (with equal weights) which points to the whole surface of the keyboard except the target and similar-to-target locations.

the initial state distribution, determines the chances of starting from each state given an observation. Figure 5b shows how we create a word model based on the HMMs of it's comprising characters. When finding a target character, we assume the transition probabilities to be proportional to the initial state distribution for the next character. Although, due to some memory effects the transition probabilities and initial state distribution might not be exactly the same, the difference seems to be negligible. Beside major reduction in training, it is only by this assumption that we are able to build a model that can accommodate unlimited number of words.

III. Information Fusion Using Token Passing

Although the experiments show that our tri-state HMM (TSHMM) can reliably be used in task inference in the eye-typing application, there are other sources of information that could be applied to the inference to improve the performance of the model. Probability distribution of task priors is a source of information that we use on a daily basis to make inferences about our observations. In our application, when the model gives us a uniform distribution over characters "V" and "U", knowing that the proceeding character was a "Q" would help us choose "U" as the eye-typed character, because that is the character that always follows "Q" in common English words.

A similar technique is used in speech processing community to improve the results of a recognizer by applying high level constraints to the character sequences [Rabiner,

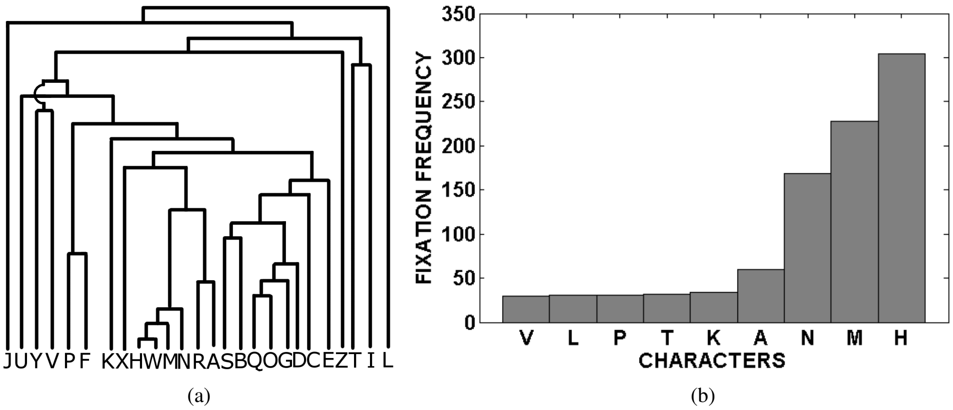


Figure 4: a) Shows the result of an experiment in perceptual measurement of image similarity that appears in [Gilmore et al., 1979, figure 1]. The results shown here are inline with what we observed in the experiments. b) Shows the top nine bars of the fixation distribution when looking for character "W". Similar characters tend to draw attention towards themselves.

1990]. When recognizing a speech signal, the constraint is imposed to the decision making engine in the form of a lexicon dictionary called *language model* (LM) that provides us with a-prior information about the current speech part that is being pronounced given the proceeding one.

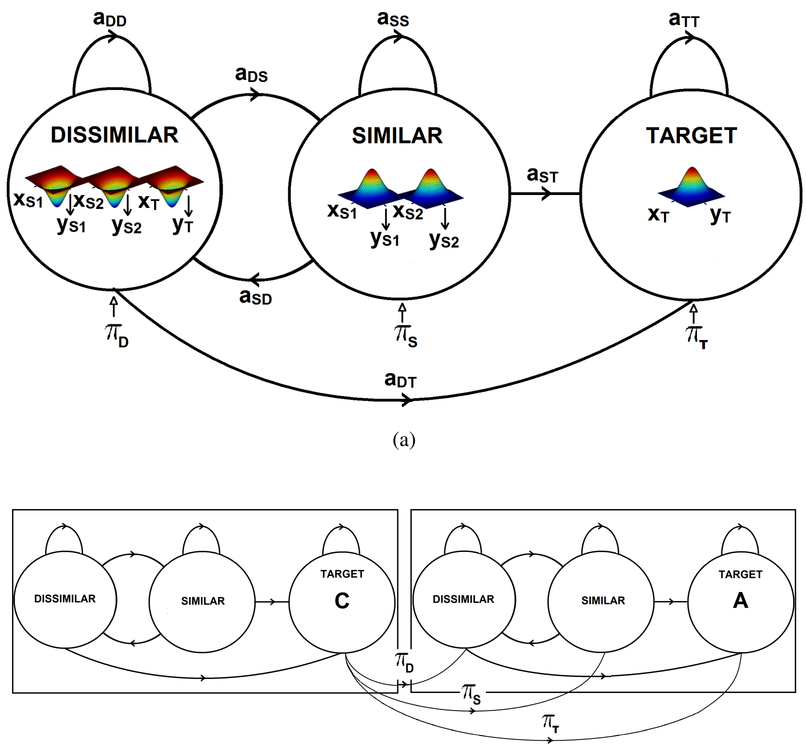
Since in our application we deal with common English words as well, we use a similar technique to apply higher order constraints on the recognizer. Depending on the order of dependency of characters, we have different orders of LM. In a *unigram* LM we assume the characters to be independent of each other. Applying a unigram LM to our TSHMM reduces the model to what we had before. However, in this paper we use a *bigram* LM whereby we impose a first order Markov process on the sequence of characters.

In order to build a LM we need to get a database of valid English words. Then we can train the bigram LM by assuming a first order Markov chain as the underlying process of character sequences. The training is done by counting the number of each pair of transitions in the corpus. In the end, a technique called *add one smoothing* is applied to the count numbers by assuming each pair occurs once more than it actually does to assign non-zero probabilities to the unseen pairs in the training corpus [Huang et al., 2001, chapter 11]. Eventually the language model gives us the probability of p_{ij} for each pair of characters (i, j), where p_{ij} is the probability of seeking character j after having found character i in our eye-typing application.

In the previous section we showed how we can train TSHMMs for each character. Therefore, by training the LM we have a complete model for the words in the dictionary that describes the transitions within the states of characters, as well as transitions between a word's characters, in a probabilistic manner. This model can be used as a generative model of the cognitive process of the human brain that generates eye movements during visual search for characters of a word (i.e., eye-typing). First we start from the initial state of a character according to the initial state distribution of the HMM, and by following the transition probabilities we can choose the states for each time step and generate observations according to the observation probabilities. When getting to the final state of a character, it is the language model that suggests which character, by what probability, can follow the current one.

The complete structure of the model for a two-character scenario is shown in figure 6. The bigram LM information is applied to the transitions between characters. Unlike the nodes inside the boxes that represent the states in the character models, the LM nodes don't represent states, which means neither any observation is generated in them nor transition through them takes up any time-step in the sequence. The LM nodes are equivalent to the so-called *grammar nodes* in the speech processing literature and is merely an indication of applying LM to the model [Huang et al., 2001, page 618].

Having the generative model of eye movements during visual search, we can use the trained parameters to decode a test eye movement trajectory to infer what character sequence has been eye-typed. If we had a limited number of hypotheses (words in the dictionary), we could use *Viterbi algorithm* to classify the test date into one of the words in the dictionary [Rabiner, 1990]. Viterbi algorithm, though, requires the word models to be built beforehand to be able to compare the likelihood of each word in the dictionary. However, for a recognition task, there might be an enormous number of words in the dictionary which makes it computationally expensive to build the word model for each word statically in the Viterbi algorithm.



(a)

(b)

Figure 5: The structure of the tri-state HMM (TSHMM) for character recognition. a) The TSHMM for a single character. b) Concatenating the character models to build up the word “CA”. The transitions between the states are governed by the initial state probabilities.

An analogous problem exists in the literature related to speech recognition, where the dictionary of possible words exceeds a certain number. The technique used there, that we propose to be used for our problem as well, is a dynamic programming algorithm called *token passing* [Young et al., 1989]. In order to find the best sequence of states that matches the observation sequence, we assign a cost R_{ij} to each transition from state i to j in an HMM equal to $\log(1/a_{ij})$ and call it the *transition cost*. a_{ij} is the transition probability in the TSHMMs model of a character if the transition is within a character model. If the transition is between characters, we use the LM statistics to evaluate the transition cost and we will have $R_{ij} = \log(1/p_{ij})$, where p_{ij} is the probability of going from character i to character j according to the LM.

The second type of cost that we use in the token passing method is called *local cost function* and defines the cost of being at state j at time t. Suppose we have data of the form $< \mathbf{Q}, y >$, where $y \in Y$ is a task label in the set of all task labels Y and $\mathbf{Q}$ is the vector containing the observation sequence of fixation locations $(\overrightarrow{q}_1, \overrightarrow{q}_2, ..., \overrightarrow{q}_T)$ sampled from a stochastic process $\{\overrightarrow{q}_t\}$ at discrete times $t = \{1, 2, ..., T\}$ over random image locations denoted in Cartesian coordinates by $\overrightarrow{q}_t = (x_t, y_t)$.

The local cost function is defined as the cost of being at state state j at time t and is defined by $L_j(t) = \log(1/b_j(\overrightarrow{q}_t))$, where $b_j(\overrightarrow{q}_t)$ is the probability of observing $\overrightarrow{q}_t$ at state j. Similar to the transition cost, L can be calculated using the parameters of the task-specific TSHMMs.

Having defined these two cost functions, we can calculate the alignment cost for an observation sequence Q and a sample state sequence $I = (i_0, i_1, ..., i_T)$ by computing the alignment cost:

$$S(I) = \sum_{\tau=1:T} (R_{i_{\tau-1} i_\tau} + L_{i_\tau}(\tau)). \tag{1}$$

However, most of the time the state sequence is hidden from the observer and therefore we can’t compute the alignment cost function directly. In token passing method, thus, we define a new alignment cost function called *local alignment cost function*, $s_j(t)$, that is equal to the sum of transition and local cost functions that leads to being at state j at time t. [1] Algorithm 1 shows how we can use the local

[1] In Viterbi algorithm (for limited number of tasks) we can use dynamic programing to calculate the alignment cost functions using the following equation:

$$s_j(t) = \min_i [s_i(t-1) + R_{ij}] + L_j(q_t). \tag{2}$$

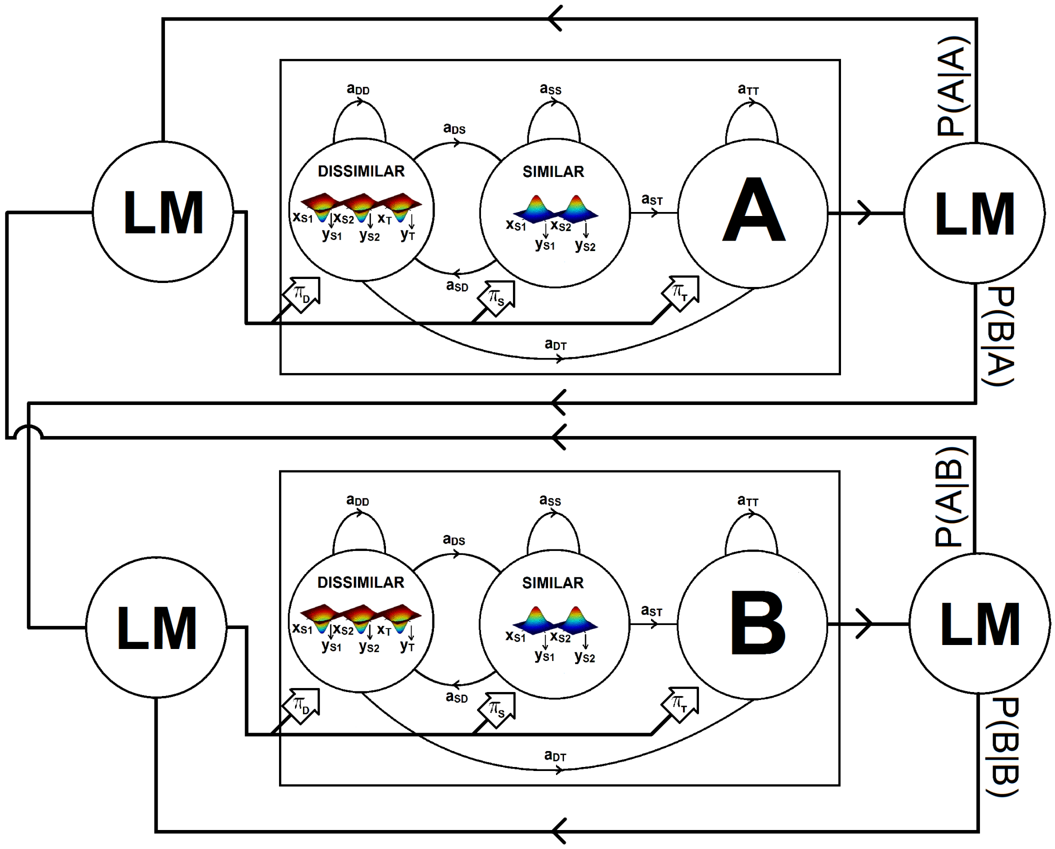


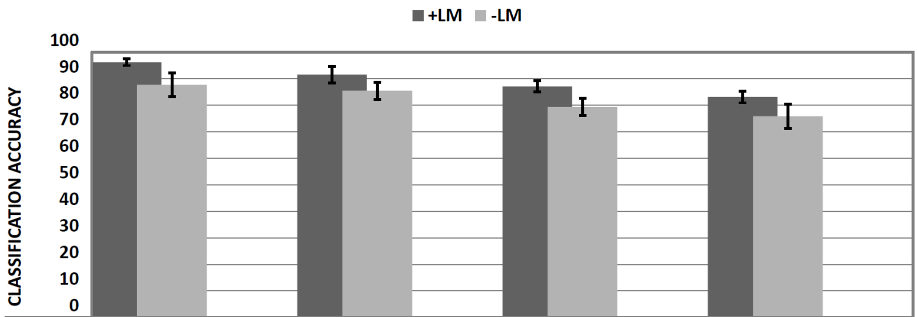
Figure 6: Incorporating the a-priori information in the form of a language model in the word model. The best state sequence for a given observation sequence can be obtained by using the token passing technique on this general word model. The LM nodes (a.k.a grammar nodes) don't generate observations, but the LM parameters are applied to the model when passing through these nodes.

alignment costs to decode a sequence of eye movements by finding the path with the minimum cost in figure 6. To do so, we assume that each HMM state can hold a movable token. We can think of a token as an object that can move from one state to another in our network. Each token carries with it a local alignment cost, which gets propagated in the network according to the transition and local cost functions. In the algorithm we refer to this cost function as the value of the token. At the end of the iterations, the rout with the minimum cost gives us the best alignment between the states and the observation sequence.

IV. Experiments

To build a database of task-dependent eye trajectories, we ran a set of trials and recorded the eye movements of six subjects while eye-typing 26 different 3-character words. The trials started with a fixation mark of size 0.26×0.26 deg appearing at the center of the screen. After foveating the fixation mark, the participant initiated the trial with a key-press. Once a trial was triggered, the word to be eye-typed was shown at the center of the display. Once the subject indicated his readiness by pressing a key, another fixation mark appeared at the center followed by an on-screen keyboard similar to the one shown in figure 2. At this phase subjects eye-typed the word by searching for the characters appearing in it as quickly as possible and signaled when they were done by pressing a key (subjects were only told to eye-type the words as quickly as possible and press a key when done). Then by asking about the location of one of the characters (selected randomly) we verified to see if the subject had correctly eye-typed the words. Once the question is answered (by fixating the right location that contained the character during the experiment and pressing a bottom) the next word is shown and the trial carries on.

The stimuli were generated by a computer and displayed on a 1280×800 pixel screen at a distance of 18 inches (1 degree of visual angle corresponds to 30 pixels, approximately). Each keyboard was composed of 25 uppercase English characters randomly located on a 5×5 grid superimposed on a gray background (we removed the letter "Z" in order to have a square layout to reduce directional



	DICT. SIZE	W026	W052	W104	W312
MEAN	+LM	96.2	91.5	87.1	83.1
	-LM	87.7	85.4	79.3	75.8
SEM	+LM	1.3	3.2	2.1	2.2
	-LM	4.4	3.3	3.2	4.6

Figure 7: Comparison of the task classification accuracy using TSHMM with a bigram LM (+LM) and TSHMM with a unigram LM (-LM) in the eye-typing application. The TSHMM with a unigram LM (-LM) corresponds to previous work, where no LM was assumed. The “DICT. SIZE” row shows the number of words (hypotheses) used in each experiment with a “Wxxx” code, where “xxx” shows the number of words. Each bar shows the mean classification rate (%) of correctly recognizing the intended word in the eye-typing application. The mean value and the standard error of the mean (SEM) are represented by bars and the numerical values are given in the following table.

Algorithm 1 Token Passing algorithm

Initialize:
Assign a zero valued token to the initial states of the models.
Assign an ∞ valued token to all other states.
Algorithm:
for t = 1 to T **do**
  for each state i **do**
    Copy the token in each state i to the connecting state j and increment its value by $R_{ij} + L_j(t)$
  end for
  Discard the original tokens.
  for each state i **do**
    Keep the token with the minimum value and discard the rest.
  end for
end for
Termination:
The token with the minimum s value in all possible final states corresponds to the best match.

bias). The 3-letter words were selected so that there was no repetition of characters in them. At the beginning of every experimental session, we calibrated the eye tracker by having the participant look at a 16-point calibration display that extended to 10×10 degrees of visual angle (the area covered by the calibration grid is stretched beyond the stimuli which spans a 6.6×6.6 degrees of visual angle).

An eye tracker (ISCAN RK-726PCI) was used to record the participant’s left eye positions at 60 Hz and a chin rest was used to minimize head movements. The eye tracker’s vertical resolution is approximately 0.11 degrees and its horizontal resolution is 0.06 degrees. An LCD monitor was used for displaying the images and the subjects used both eyes to conduct the experiments.

After recording eye movements, data analysis was carried out on each trial wherein we removed the blinks, outliers and trials with wrong answers in the verification phase from the data and classified the eye movement data into saccades and fixations. Moreover, in some of the initial trials, after eye-typing the word, the viewer returned to the locations of the characters to double-check the coordinates of them. In order to simulate a real eye-typing application we removed these parts from the trajectories in the pre-processing as well.

After the preprocessing we obtained a database of 145 trajectories of the form ($\vec{q}_1, \ldots, \vec{q}_T$), each containing observation sequences of coordinates of fixations while performing the eye-typing, where $\vec{q}_t = (x_t, y_t)$ represents x-coordinate and y-coordinate of the t^{th} fixation, respectively.

In order to perform the evaluation, we compare the results of our proposed model that uses a TSHMM and a bigram LM to model the tasks, with the one proposed in [Haji-Abolhassani and Clark, 2012b], that uses a TSHMM with no LM (i.e., with a unigram LM), in four different dictionary sizes. We denote the TSHMM that uses the lexicon information by *+LM* and the TSHMM that disregards any high-level information by *-LM*. We created four sets of dictionaries of 26, 52, 104 and 312 English words using the Carnegie Mellon pronouncing dictionary (CMPD) [Weide, 2005]. All dictionaries were built so that they all include all the words of the smaller dictionaries. The words were selected randomly from the CMPD and the words length varied between three to five characters. The language model was also created using the CMU-Cambridge toolkit [Clarkson and Rosenfeld, 1997] by extracting language models from the words in dictionaries.

In order to train the TSHMMs, we have to adjust the mean vector of the 2-D Gaussians according to the training character so that it aligns with the center of character location. According to [Rabiner, 1990] a uniform (or random) initial estimation of initial state and transition probabilities (Π and A) is adequate for giving useful re-estimation of these parameters (subject to the stochastic and the non-zero value constraints). Thus, we set a random initial values for the parameters in the generic HMM and run the *Baum-Welch* algorithm on the training set to obtain the final TSHMM [Huang et al., 1990]. We also used a technique called parameter tieing [Rabiner, 1990] to force a unique task and stimuli independent covariance matrix across all of the Gaussian distributions in the mixtures. Thus, we can build the word model for the test data by dynamically changing the means of the states according to the character locations of the characters and using the estimated variances of characters.

Figure 7 shows the accuracy of word inference using TSHMM with LM (+LM) and TSHMM without LM (-LM) methods ranging over four dictionary sizes. As expected, the +LM performs better than -LM due to the fusion of information provided by the LM. The table below the figure shows the accuracy and the standard error of the mean (SEM) of the corresponding bars. For each bar we ran a 10-fold cross validation on our database of 145 trajectories in order to define the training and test sets and used the same epochs across all the methods.

Acknowledgment

The authors would like to thank *Fonds de recherche du Qubec - Nature et technologies* (FQRNT) and *Natural Sciences and Engineering Research Council of Canada* (NSERC) for supporting this work.

References

G.T. Buswell. *How people look at pictures: A study of the psychology of perception in art*. Chicago: University of Chicago Press, 1935.

M.S. Castelhano, M.L. Mack, and J.M. Henderson. Viewing task influences eye movement control during active scene perception. *Journal of Vision*, 9(3):6, 2009.

J.J. Clark and J.K. O'Regan. Word ambiguity and the optimal viewing position in reading. *Vision Research*, 39 (4):843–857, 1998.

P. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Fifth European Conference on Speech Communication and Technology*, 1997.

B. Fischer and H. Weber. Express saccades and visual attention. *Behavioral and Brain Sciences*, 16:553–553, 1993.

GC Gilmore, H. Hersh, A. Caramazza, and J. Griffin. Multidimensional letter similarity derived from recognition errors. *Attention, Perception, & Psychophysics*, 25(5): 425–431, 1979.

A. Haji-Abolhassani and J.J. Clark. Realization of an inverse yarbus process via hidden markov models for visual-task inference. *Journal of Vision*, 11(11):218–218, 2011a.

A. Haji-Abolhassani and J.J. Clark. Visual task inference using hidden markov models. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2011b.

A. Haji-Abolhassani and J.J. Clark. A computational model for visual-task inference in pop-out and conjunction search. *submitted to Vision Research*, 2012a.

A. Haji-Abolhassani and J.J. Clark. Visual task inference in conjunction search using hidden markov models. *accepted for presentation in Vision Sciences Society 2012 - to appear in Journal of Vision*, 2012b.

H.V. Helmholtz. Handbuch der Physiologischen Optik, Dritter Abschnitt, 1896.

J. Hu, M.K. Brown, and W. Turin. HMM based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.

X. Huang, A. Acero, H.W. Hon, et al. *Spoken language processing*. Prentice Hall PTR New Jersey, 2001.

X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

G. Keren and S. Baggen. Recognition models of alphanumeric characters. *Attention, Perception, & Psychophysics*, 29(3):234–246, 1981.

V. Nair and J.J. Clark. Automated visual surveillance using hidden markov models. In *International Conference on Vision Interface*, pages 88–93, 2002.

L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296, 1990.

A.M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980. ISSN 0010-0285.

R. Weide. The carnegie mellon pronouncing dictionary [cmudict. 0.6], 2005.

A.L. Yarbus. Eye movements during perception of complex objects. *Eye movements and vision*, 7:171–196, 1967.

S.J. Young, NH Russell, and JHS Thornton. Token passing: a simple conceptual model for connected speech recognition systems. *Cambridge University Engineering Department*, pages 1–23, 1989.