Commonsense on the go

H Lieberman, A Faaborg, J Espinosa and T Stocky

Mobile devices such as cell-phones and PDAs present unique opportunities and challenges. Naïve transfer of applications from full-size computers often fails because the interaction becomes too cumbersome and challenging for the user. We address these opportunities and challenges by giving portable devices commonsense knowledge — a large collection of simple facts about people and everyday life. We illustrate our approach with descriptions of several applications we have implemented for mobile devices using the Open Mind and ConceptNet resources. These include a dynamic phrasebook for tourists, an assistant for searching personal social networks, and a predictive typing aid that uses semantic information rather than statistics to suggest word completions.

1. Introduction

Computers lack commonsense. Current software applications know literally nothing about human existence. Because of this, the extent to which an application understands its user is restricted to simplistic preferences and settings that must be directly manipulated. Current mobile devices are very good at following explicit directions (like a cell-phone that does not ring when set to silent), but are completely incapable of any deeper level of understanding or reasoning.

Once mobile devices are given access to commonsense knowledge, millions of facts about the world we live in, they can begin to employ this knowledge in useful and intelligent ways. Mobile devices can understand the context of a user's current situation and what is likely to be going on around them. They can know that if the user says 'my dog is sick' they probably need a veterinarian; and that tennis is similar to basketball in that they are both physical activities that involve athletes and give people exercise. Mobile devices will be able to understand what the user is trying to write in a text message and predict what words they are trying to type based on semantic context. In this paper we will demonstrate mobile applications that use commonsense knowledge to do all of these things. This approach enables new types of interactions with mobile devices, allowing them to understand the semantic context of situations and statements, and then act on this information.

1.1 Teaching computers the stuff we all know

Since the fall of 2000 the MIT Media Lab has been collecting commonsense facts from the general public through a Web site called Open Mind [1—3]. At the time of this writing, the Open Mind Common Sense project has collected over 700 000 facts from over 14 000 participants. These facts are submitted by users as natural language statements of the form 'tennis is a sport' and 'playing tennis requires a tennis racket'. While Open Mind does not contain a complete set of all the common sense knowledge found in the world, its knowledge base is sufficiently large enough to be useful in real-world applications.

mobile devices will employ commonsense to understand the user's context

Using natural language processing, the Open Mind knowledge base was mined to create ConceptNet [4], a large-scale semantic network currently containing over 300 000 nodes. ConceptNet consists of machine-readable logical predicates of the form: [IsA 'tennis' 'sport'] and [EventForGoalEvent 'play tennis' 'have racket']. ConceptNet is similar to WordNet [5] in that it is a large semantic network of concepts; however, ConceptNet contains everyday knowledge about the world, while WordNet follows a more formal and taxonomic structure.

For instance, WordNet would identify a dog as a type of canine, which is a type of carnivore, which is a kind of placental mammal. ConceptNet identifies a dog as a type of pet [4]. For more information about the creation and structure of ConceptNet, see Liu and Singh [4].

We have leveraged the knowledge of human existence contained in ConceptNet to create three intelligent mobile applications — a dynamic phrasebook for tourists [6], a match making agent for searching your local social network [7], and a new approach to predictive text entry [8, 9].

2. Using commonsense reasoning to create a dynamic phrasebook for tourists

When travelling in foreign countries, people often rely on traditional phrase books for language translation. However, these phrase books only work in a limited number of common situations, and even common situations will often deviate from the predefined script on which the phrase book relies. Translation software exists for personal digital assistant (PDA) devices, but users must write out every phrase they wish to translate, slowing communication. We aim to solve both problems with a mobile application called GloBuddy 2. Using ConceptNet and Open Mind, GloBuddy 2 is able to expand on the user's translation request and provide words and phrases related to the user's situation. The result is a dynamic phrase book that can respond to the user's particular situation due to its breadth of commonsense knowledge about the world. GloBuddy 2 is often more effective than using a conventional phrase book because it contains broad knowledge about a wide variety of situations.

2.1 Introduction

Communication between two people who do not speak the same language is often a difficult and slow process. Phrase translation books provide contextually relevant information, but can only cover a limited set of extremely common situations. Dictionaries can translate a wide range of words, but are very slow to access. The same is true with PDA-based translation software. While it is considerably faster than looking up each word in a physical book, writing each phrase into the device is still a tedious and time-consuming task. The best solution is to use a human translator, someone who is capable of going beyond simply translating your words and can intelligently understand their context. A human translator would know to ask, 'where can I find a doctor' if you were ill or to ask, 'where is a restaurant' if you were hungry. A human translator knows that you can find a location using a map, you can get to a location using a taxi, and that when you arrive you should tip the driver. A human translator is the best solution, not just because phrases are translated quickly, but rather because they can use commonsense reasoning to expand upon your initial request.

We have been able to implement this type of commonsense reasoning into a mobile language translation agent called GloBuddy 2. GloBuddy 2 uses Open Mind [1—3], and ConceptNet [4] to understand its user's situation.

2.2 User interface

When launching GloBuddy 2, the user is provided with two modes — interpreting a statement in a foreign language, and preparing to say a statement in a foreign language. They can also select which language they would like to use.

In our testing, English-speaking users have had some difficulty typing statements said to them in a foreign language. We are now investigating several solutions to this problem, including speech recognition and allowing users to write in phrases phonetically to the device. However, we are still in the early stages of testing these approaches. In preliminary testing we have found that this problem is not as significant when dealing with more phonetic languages like Spanish and Italian.

Where GloBuddy 2 differs from traditional translation applications is in the way it translates the user's statements into a foreign language. In addition to directly translating what the user types, GloBuddy 2 uses Open Mind and ConceptNet to expand on the user's translation request, providing them with a localised vocabulary of related terms and phrases.

While the user can enter a complete phrase for translation, GloBuddy 2 only needs a few words to begin finding relevant information. After the user enters a phrase or a set of concepts, GloBuddy 2 prepares contextually relevant information. First, GloBuddy 2 translates the text itself. It then extracts the key concepts the user entered, and uses ConceptNet to find contextually related words and the Open Mind knowledge base to find contextually related phrases. After performing this commonsense reasoning, GloBuddy 2 then displays all of this information to the user. For instance, if the user enters the term picnic, GloBuddy 2 expands on the term, as shown in Fig 1.

GloBuddy 2 uses common sense to expand on a user's translation request

Even by entering only one word, the user is given a pre-translated localised vocabulary of terms that they may find useful in their current situation.

2.3 User scenario

To demonstrate GloBuddy 2's functionality (see Fig 2), consider a hypothetical scenario. While bicycling through France, our non-French speaking user is injured in a bicycle accident. A person approaches and asks 'Avez vous besoin d'aide?' The user launches GloBuddy 2 on their Pocket PC and translates this statement to 'Do you need assistance?' The user has two goals:

- find all the parts of their now demolished bicycle,
- get medical attention.

The user quickly writes three words into GloBuddy 2 to describe their situation — doctor, bicycle, accident.

In the related words category, accident expands to terms like unintentional, mistake and costly. The term doctor expands to terms like hospital, sick, patient, clipboard, and medical attention. And bicycle expands to pedal, tyre, seat, metal, handle, spoke, chain, brake, and wheel. By quickly writing three words, the user now has a localised vocabulary of pre-translated terms to use in conversation.

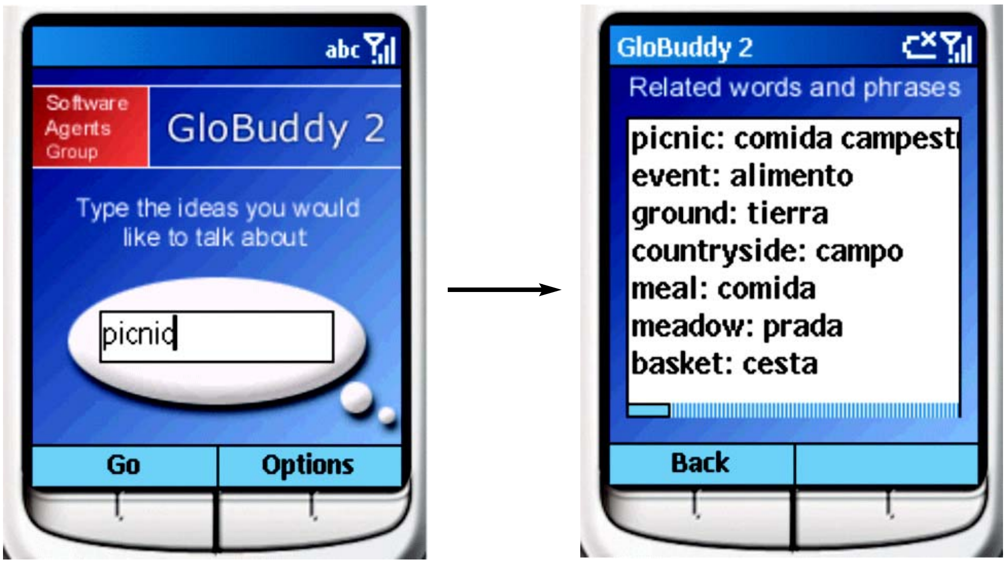

Fig 1 A localised vocabulary expanding 'picnic'.

t is important to note that not all of these words and phrases returned by GloBuddy 2 are guaranteed to be particularly relevant to the user's exact situation. For instance, clipboard (returned because it is held by a doctor and contains medical information) and veterinarian (also returned because of the relationship with the concept doctor) are particularly irrelevant, as is human. Often relevance depends on the exact details of the user's situation. While the commonsense reasoning being performed by GloBuddy 2 is not perfect, it is good enough to reasonably expand upon the user's input for an extremely broad range of scenarios.

By directly searching the Open Mind knowledge base, GloBuddy 2 also returns complete phrases that may relate to the user's situation, shown in Fig 3. The phrases are run through the Babel Fish translator [10], so translations are not always exact. Additionally, while these phrases may seem somewhat awkward in conversation (like the phrase in the example in Fig 3, which is in third person) they are good enough to get the point across. GloBuddy 2 is most effective when used by people who have enough knowledge of a foreign language to rearrange sentences, but lack a large vocabulary.

From this example we can see the advantages of using commonsense reasoning in a language translation device:

- users do not have to write the entire statement they wish to say, resulting in faster communication,
- GloBuddy 2 is able to find additional concepts that are relevant to users' situations,
- GloBuddy 2 is able to provide users with complete phrases based on concepts they entered.

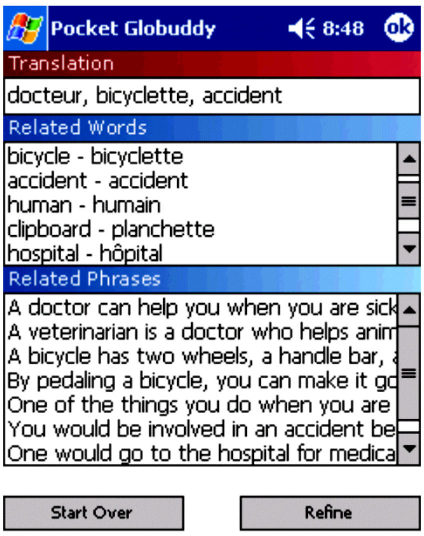

Fig 2 The user relies on GloBuddy 2 to describe their bicycle accident.

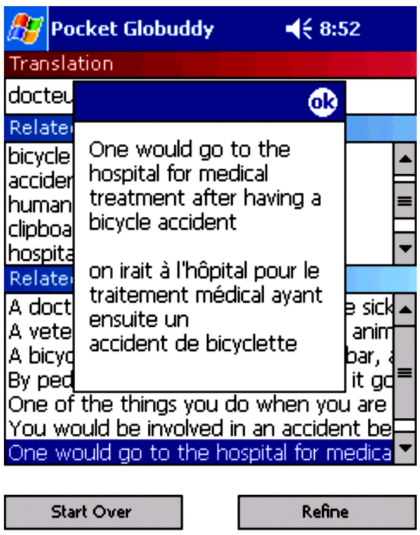

Fig 3 GloBuddy 2 returns phrases from Open Mind.

y only writing three words and tapping the screen twice, our injured bicycle rider was able to say phrases like 'on irait á l'hôpital pour le traitement médical ayant ensuite un accident de bicyclette', and had access to many additional words and phrase.

2.4 Implementation

The first version of GloBuddy [11] was implemented as a software application for laptop computers. GloBuddy 2 has been implemented and tested on the Microsoft PocketPC and Smartphone platform's using C# and the .NET Compact Framework, and on the Nokia 6600 using the Java 2 Micro Edition (J2ME).

Currently GloBuddy 2 is implemented using a thin client architecture. Open Mind and ConceptNet are accessed over the Internet using Web Services. Translation is completed using a Web service interface to AltaVista's Babel Fish [10].

2.5 Evaluation

To determine GloBuddy 2's effectiveness as a language translation aid in a wide range of environments and social settings, we evaluated both GloBuddy 2's ability to use commonsense reasoning to make inferences that were contextually relevant to the user's situation, and also GloBuddy 2's design and user interface.

2.6 Evaluation of GloBuddy 2's knowledge base

To evaluate the general quality of words and phrases GloBuddy 2 returns, we asked users to generate a set of 100 unique situations that people travelling in foreign countries could find themselves in. We then tested GloBuddy 2's ability to find relevant words and phrases for each particular situation, recording the number of contextually accurate concepts returned. For instance, in the situation of being arrested, GloBuddy 2 was able to expand the single concept of arrest, to the concepts of convict, suspect, crime, criminal, prison, jury, sentence, guilty, appeal, higher court, law, and accuser.

We found that when given a single concept to describe a situation, GloBuddy 2 was able to provide users with an average of six additional contextually relevant concepts for use in conversation.

2.7 Evaluation of GloBuddy 2's user interface

In a preliminary evaluation of the design of GloBuddy 2, we studied four non-Spanish speaking users as they tried to communicate with a person in Spanish. For each scenario, the users alternated between using GloBuddy 2, and a Berlitz phrase book with a small dictionary [12]. The experiment was video taped, and after completing the scenarios the users were interviewed about their experience. We found that for a stereotypical situation like ordering a meal in a restaurant, while GloBuddy 2 provided a reasonable amount of information, the Berlitz phrase book was more useful. However, when attempting to plan a picnic, users had little success with the phrase book. This is because the task of planning a picnic fell outside the phrase book's limited breadth of information. Users found GloBuddy 2 to be significantly more useful for this task, as it provided contextually relevant concepts like basket, countryside, meadow and park.

Using GloBuddy 2 still resulted in slow and deliberate conversations due to network lag and the speed at which users could enter words into the PDA. However, GloBuddy 2's ability to retrieve contextually related concepts reduced both the number of translation requests and the amount of text entry. This made conversations more fluid compared to using a traditional PDA dictionary.

2.8 Discussion — breadth-first versus depth-first approaches to translation

GloBuddy 2 performed noticeably better than a traditional phrase book for uncommon tasks in our evaluations. To understand why, consider the knowledge contained in a phrase book, a translation dictionary, and a human translator. In Fig 4 we see that there is usually a trade-off between a system's breadth of knowledge, and its depth of reasoning.

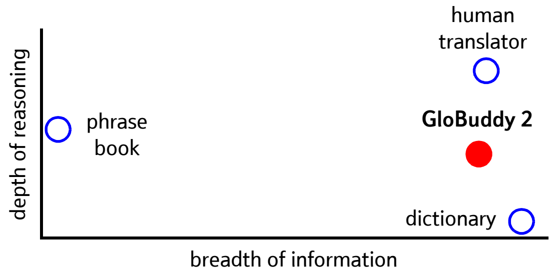

Fig 4 The trade-off between a system's breadth of information and its depth of reasoning.

A phrase book can provide a deep amount of information about a small number of stereotypical tourist activities, like checking into a hotel. At the other end of the spectrum, a translation dictionary provides a much broader set of information, but has effectively no depth, as it provides the user with only words and their specific definitions. The best solution between these two extremes is a human translator. However, GloBuddy 2 is able to break this traditional trade-off by accessing a vast amount of commonsense knowledge that humans have entered into Open Mind.

GloBuddy 2 is unique in that it provides a significant breadth of information along with a shallow amount of reasoning. While GloBuddy 2 does not contain the same level of depth as a phrase book, it can provide commonsense reasoning over a much broader realm of information and situations.

2.9 The need for a fail-soft design

GloBuddy 2 makes mistakes. This is partly because almost all of the commonsense facts in Open Mind have obscure exceptions, and also because accurate commonsense reasoning can be of little consequence to the user's particular situation. For instance, if a user has just been injured and is interested in finding a doctor the concept of clipboard is not particularly important. However, if the user has arrived at the hospital and a confused nurse is about to administer medication, the user may be happy to see that GloBuddy 2 returned the concept.

Aside from using up screen space, the incorrect inferences that GloBuddy 2 makes are of little consequence. They do not crash the software, significantly confuse the user, or significantly reduce the overall effectiveness of the device. This type of fail-soft design is important when creating software that algorithmically reasons about the imprecise realm of everyday human activities.

2.10 Future work

In the near future we will be updating GloBuddy 2 so that it will not require an Internet connection, but will instead access commonsense knowledge and translations from a 512 MB external storage card.

A future version of GloBuddy may include the ability to perform temporal reasoning, prompting users with translations based on previous requests. While ConceptNet does not explicitly include the information needed to make these types of temporal inferences, LifeNet [13, 14] contains these types of cause and effect relationships.

Ideally, future versions of GloBuddy will use speech recognition and generation, further reducing input and facilitating more fluid conversations.

2.11 Conclusion — using commonsense reasoning to understand the user's situation

The majority of Smartphone and PDA applications fail to take advantage of the fact that people use them in a variety of situations. To create an application that understands the context of the user's surroundings it must have access to a large knowledge base of commonsense facts. GloBuddy 2 is a good example of how mobile applications can leverage commonsense knowledge to understand and respond to the user's particular situation. However, this is only one example of leveraging this information. Commonsense knowledge has also been effectively used to identify the topics a user is talking about by listening to their conversations [15]. Beyond understanding a user's situation, commonsense knowledge can also be used to understand a user's underlying goals. This is demonstrated in our next application.

3. Using commonsense reasoning to improve searching social networks

Despite their inherently social purpose, the increased processing power and network connectivity in modern cell-phones is rarely utilised for social applications. Modern processors, higher resolution screens and increased memory have been mainly utilised by games. And aside from text messaging, the network bandwidth available to telephones is often used for solitary tasks like reading horoscopes and news stories. We have developed a cell-phone-based application that uses the device's processing power and network connectivity for a social purpose: to allow users to perform real-time searches on their local social network, against pieces of information that their contacts have provided about themselves. The system we have designed is similar to Expert Finder [16], a software agent to help novices find experts in a particular domain, and Friendster [17] a Web site that uses social networks for the purposes of dating and meeting new people. However, unlike these systems, our match making agent uses commonsense reasoning to understand user's goals and for query expansion based on analogies.

3.1 User interface

Users can access their profile through a Web site and manage their personal information and privacy settings. Here users can enter statements about their interests and activities. Users can then search against their contact's information and one level beyond in their social network using a cell-phone-based application shown in Fig 5.

The system uses commonsense reasoning algorithms when processing searches to:

- expand upon the user's query to contextually similar topics if it cannot find a direct match,
- allow the user to enter goal-based searches.

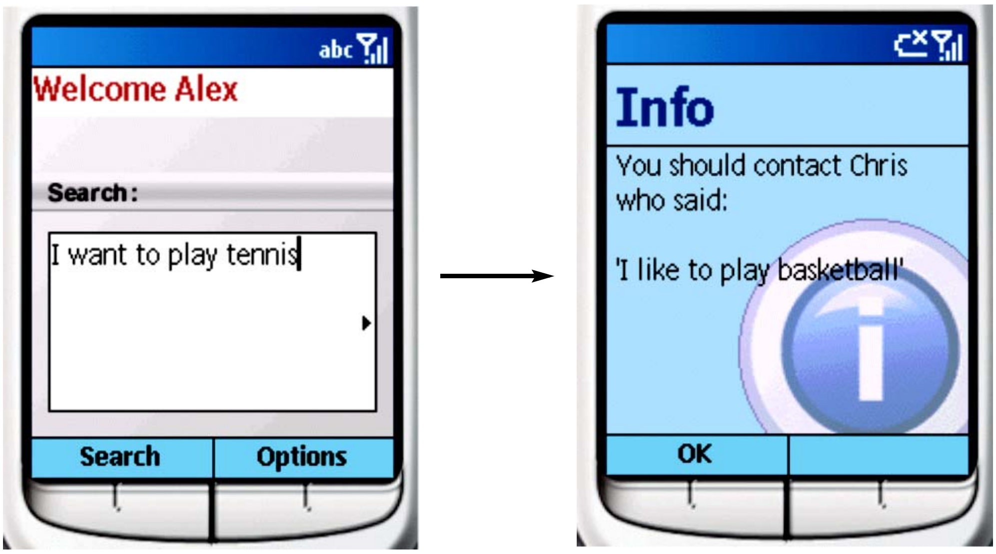

Fig 5 Searches are expanded to analogous concepts.

Both of these problems are solved using ConceptNet [4].

A problem facing text searches over a limited number of profiles is that the probability of direct matches is low. To deal with this problem, our application uses commonsense reasoning to expand on the user's query by making analogies. For instance, if a user enters the search 'I want to play tennis', and the system is not able to find a direct match on the term tennis, it can then expand the search to analogous concepts.

A user's search for playing tennis might return the result of someone playing basketball. While this is not a direct match, the two concepts share a number of links in ConceptNet, as shown in Fig 6.

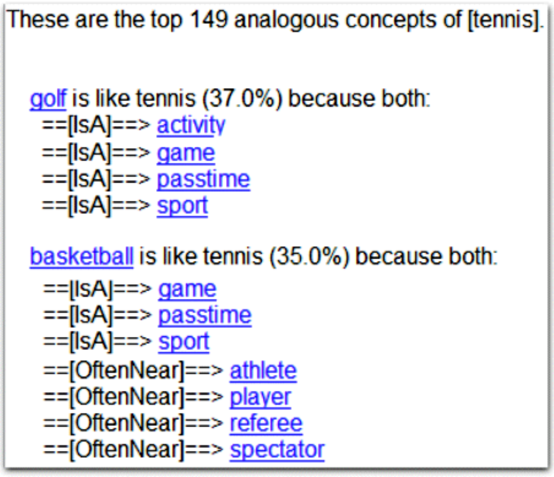

Fig 6 Using conceptual links to find analogous concepts.

While the user may not be interested in playing basketball instead of tennis, it is possible that they simply had the higher-level goal of playing a sport.

A second problem facing text searches is that novice users will often enter goal-based statements that cannot be resolved with simple keyword matching. For instance a user might type 'my dog is sick' rather than 'I need to find a good veterinarian'. We are currently adapting our system to employ a technique that has been previously used to process goal-based Web searches in a project called GOOSE [18]. This is achieved by parsing the query into a semantic frame and then a commonsense sub-domain. Then, the query is reformulated though commonsense inference guided by expertise templates [18].

3.2 Conclusion — using commonsense reasoning to understand user's goals

By using ConceptNet [4] to understand user's goals, this application is able to go beyond direct keyword matching and logically expand the user's search. The last two applications have used commonsense reasoning to focus on some of the opportunities of mobile devices, leveraging the fact that people use them in a variety of situations and providing just-in-time information. Our third application focuses on one of the challenges of mobile devices — text entry.

4. Using commonsense reasoning to improve predictive text entry

People cannot type as fast as they think. As a result, they have been forced to cope with the frustration of slow communication, particularly in mobile devices. In the case of text entry on mobile phones, for example, users typically have only twelve input keys, so that to simply write 'hello' requires thirteen key taps.

Predictive typing aids have shown some success, particularly when combined with algorithms that can disambiguate words based on single-tap entry. Past approaches to predictive text entry have applied text compression methods (e.g. Witten et al [19]), taking advantage of the high level of repetition in language.

Similar approaches have applied various other statistical models, such as low-order word *n*-grams, where the probability of a word appearing is based on the $n - 1$ words preceding it. Inherently, the success of such models depends on their training set corpora, but the focus has largely been on the statistics rather than the knowledge base on which they rely.

commonsense is introduced as a complement to statistical methods

We have chosen to focus on the knowledge base issue, and propose an alternative approach based on commonsense reasoning. This approach performs on a par with statistical methods and is able to anticipate words that could not be predicted using statistics alone. We introduce this commonsense approach to predictive text entry not as a substitute to statistical methods, but as a complement. As words predicted by the commonsense system tend to differ from those predicted by statistical methods, combining these approaches could achieve superior results to the individual performance of either.

4.1 Related work

Efforts to increase the speed of text entry fall into two primary categories:

- new means of input, which increase efficiency by lessening the physical constraints of entering text,
- predictive typing aids, which decrease the amount of typing necessary by predicting completed words from a few typed letters.

4.2 Means of input

Augmented keyboards have shown improvements in efficiency — both physical [20] and virtual [21] keyboards. In cases where the keyboard is constrained to a less efficient layout, disambiguation algorithms have demonstrated success in increasing efficiency [22].

Others have looked at alternate modalities, such as speech and pen gesture. Such modalities are limited by similar physical constraints to keyboard entry. And while speech recognition technology continues to improve, it is currently less efficient and less 'natural' than keyboard entry [23].

Reducing the physical constraints around entering text is extremely valuable, and we view predictive typing aids as a means to solving another part of the problem.

4.3 Predictive typing aids

One of the first predictive typing aids was the Reactive Keyboard [24], which made use of text compression methods [19] to suggest completions. This approach was statistically driven, as have been virtually all of the predictive models developed since then. Statistical methods generally suggest words based on:

- frequency, either in the context of relevant corpora or what the user has typed in the past,
- recency, where suggested words are those the user has most recently typed.

Such approaches reduce keystrokes and increase efficiency, but they make mistakes. Even with the best possible language models, these methods are limited by their ability to represent language statistically. In contrast, by using commonsense knowledge to generate words that are semantically related to what is being typed, text can be accurately predicted where statistical methods fail.

4.4 Predicting text using common sense

Commonsense reasoning has previously demonstrated its ability to accurately classify conversation topics [15]. Using similar methods, we have designed a predictive typing aid that suggests word completions that make sense in the context of what the user is writing.

4.5 Open Mind Common Sense

Our system's source of commonsense knowledge is ConceptNet [4], which is derived from Open Mind [1—3].

It would be reasonable to substitute an *n*-gram model or some other statistical method to convert Open Mind into relationships among words; the key is starting from a corpus focused on commonsense knowledge.

4.6 Using ConceptNet to complete words

As the user types, the system queries ConceptNet for the semantic context of each completed word, disregarding common stop words. ConceptNet returns the context as a list of phrases, each phrase containing one or more words, listing first those concepts more closely related to the queried word. As the system proceeds down the list, each word is assigned a score:

$$\text{score} = \frac{1}{\log_5(5+n)}$$

The variable n increments as the system works through the phrases in the context, so that the word itself ($n = 0$) receives a score of 1.0, the words in the first phrase ($n = 1$) receive a score of 0.90, those in the second phrase 0.83, and so on.

Base 5 was selected for the logarithm as it produced the best results through trial-and-error. A higher base gives too much emphasis to less relevant phrases, while a lower base undervalues too many related phrases.

The scored words are added to a hash table of potential word beginnings (various letter combinations) and completed words, along with the words' associated total scores. The total score for a word is equal to the sum of that word's individual scores over all appearances in semantic contexts for past queries. As the user begins to type a word, the suggested completion is the word in the hash table with the highest total score that starts with the typed letters.

In this way, words that appear multiple times in past words' semantic contexts will have higher total scores. As the user shifts topics, the highest scored words progressively get replaced by the most common words in subsequent contexts.

4.7 Evaluation

We evaluated this approach against the traditional frequency and recency statistical methods. Our evaluation had four conditions:

- language frequency, which always suggested the 5000 most common words in the English language (as determined by Zeno et al [25]),
- user frequency, which suggested the words most frequently typed by the user,
- recency, which suggested the words most recently typed by the user,
- commonsense, which employed the method described in the previous section.

These conditions were evaluated first over a corpus of e-mails sent by a single user, and then over topic-specific corpora.

Each condition's predicted words were compared with those that actually appeared. Each predicted word was based on the first three letters typed of a new word. A word was considered correctly predicted if the condition's first suggested word was exactly equal to the completed word. Only words four or more letters long were considered, since the predictions were based on the first three letters. The accuracy of these conditions is displayed in Fig 7.

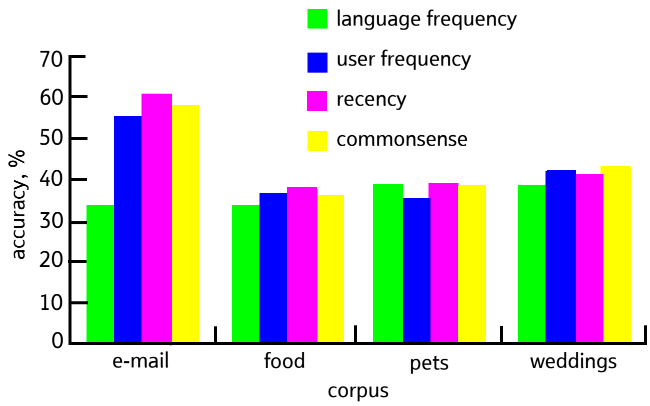

Fig 7 Accuracy of four conditions across various corpora.

4.8 E-mail corpus

As predictive text entry is especially useful in mobile devices, we compiled an initial corpus that best approximated typical messaging on mobile devices. This initial corpus consisted of a single user's sent e-mails over the past year. We decided to test our approach on e-mail messages instead of instant messaging logs since synchronous conversations often rapidly switch between topics. SMS messages would have been acceptable, given some modifications. Due to their character limit, these messages often contain extremely abbreviated words ('go 2 cnma 2nite?' converts to 'go to cinema tonight?'). These abbreviated words would need to be converted both when querying ConceptNet to establish the message's context, and when determining if our approach correctly predicted a word.

We used e-mails from only one user so that the corpus would be more suitable for the user frequency and recency conditions. There were 5500 e-mails in total, consisting of 1.1M words, 0.6M of which were four or more letters long.

The results showed that 'recency' performed best, with an overall accuracy of 60.9%, followed by commonsense at 57.7%, user frequency at 55.1% and language frequency at 33.4%.

Overall, the performance of the commonsense approach was on a par with the other conditions. Upon further analysis, it became clear that our system performed better relative to the other conditions when there was better coverage of the current topic in ConceptNet. Many of the e-mails were rather technical in nature, on topics scarcely mentioned in the commonsense database. By ConceptNet's very nature, its broad knowledgebase is not evenly distributed over all topics, so some topics experience more in-depth coverage than others.

With this in mind, we evaluated the four conditions on three additional corpora, which represented areas where ConceptNet had fairly significant coverage.

4.9 Topic-specific corpora

Evaluation was run over three additional corpora representing topics covered fairly well by ConceptNet:

- food: 20 articles from Cooking.com, selected at random — 10 500 total words, of which 6500 were four or more letters long,
- pets: 20 articles from PetLifeWeb.com, selected at random — 10 500 total words, of which 6000 were four or more letters long,
- weddings: 20 articles from WeddingChannel.com, selected at random — 16 500 total words, of which 10 000 were four or more letters long.

The results (summarised in Fig 7) showed once again that the commonsense approach was on a par with the other conditions, performing best on the weddings corpus, where, of the three corpora, ConceptNet has the best coverage.

4.10 Where the commonsense approach excels

Once again, we completed a detailed analysis of where the commonsense approach performed best and worst relative to the other conditions. Our system performed best (as much as 11.5% better on a 200 word section than the next best method) in cases of low word repetition, especially at times when the words selected were somewhat uncommon, as judged by the words' ranking in Zeno et al [25].

The following excerpt from the data illustrates this point:

> 'I spoke to my roommate — sorry the rent isn't on time, he said he did pay it right at the end of last month'

In this case, there are several words that the commonsense system is able to predict correctly, while the others are not. Based on two of the first words typed — 'spoke' and 'roommate' — the system predicts three of the words that follow — 'rent', 'time', and 'right'. Those words, in turn, allow the prediction of 'last' and 'month'. In total, of the last eight words four or more letters long, the commonsense system correctly predicts six (75%) of them, based only on two typed words and the predicted words themselves.

4.11 Implementation

The commonsense predictive text entry system was originally implemented on the Java 2 Platform, Standard Edition (J2SE), making use of the ConceptNet Java API.

Similar versions were implemented on a Motorola MPx200 Smartphone (see Fig 8) and a Pocket PC, using C# with the .NET Compact Framework, as well as on a Nokia 6600, using the Java 2 Platform, Micro Edition (J2ME) with MIDP (Mobile Information Device Profile) 1.0. Due to memory constraints, these versions used a subset of ConceptNet — approximately 80 000 nodes. Next generation devices will not have such memory constraints, and current constraints can be overcome with the use of external memory cards.

The system serves as a predictive typing aid that predicts word completions. Once the user has typed a two-letter word beginning, the system suggests the most relevant completed word. The user can then accept that suggestion, or can

Fig 8 Screenshot of the Smartphone implementation.

continue typing, which may result in a new predicted word completion based on the new letters.

These mobile device implementations demonstrate the feasibility of applying a commonsense system to just about any computing environment.

4.12 Comparing commonsense reasoning to n-grams

A common statistical approach to word completion is low-order word *n*-grams, where the probability of a word appearing is based on the *n*–1 words preceding it. As previously mentioned, the first difference between our approach and *n*-grams is the corpus being used. In our case the corpus consists of commonsense knowledge. While it is certainly possible to use an *n*-grams approach on a training corpus of commonsense knowledge, there are also many differences in how these two approaches function. The *n*-grams approach is a statistical technique based on frequency of adjacent words. Because *n* is usually 2 or 3 (referred to as bigrams and trigrams), this approach cannot take into account the context of a text message beyond a three word horizon. To take into account more of a message's context, *n* would need to be increased to a larger number like 10. However, this results in intractable memory requirements. Given a training corpus containing m words, the *n*-grams approach where *n* equals 10 would require storing a look-up table with (in the worst case) m^{10} entries. The *n*-gram approach is usually trained on a corpus where *m* is in the millions. Commonsense reasoning is able to escape these intractable memory requirements because:

- our training corpus is smaller,
- our parser does more natural language processing.

To look at an example, consider the text message: 'Buy me a ticket to the movie, I'll meet you at the th…'. The commonsense reasoning approach notices the words ticket and movie, and uses ConceptNet to look up the context of these two words (which returns list of about 20 words), it then concludes that 'th' completes to theatre. An *n*-grams approach would need to look up what the most likely word to start with 'th' is in all other instances of this sentence.

4.13 Future work

It is clear that commonsense knowledge is useful for predictive typing aids. While the system's performance is on a par with statistical methods, what is more important is that the words predicted using common sense differ significantly from the other conditions. This suggests that the question is therefore not which method to use but how to combine the methods effectively, to exceed the performance of any individual method.

4.14 Combining commonsense and statistical methods

One technique for combining commonsense and statistical methods would be to treat the contributions of each individual approach as multiple hypotheses. These hypotheses could then be weighted based on user behaviour, as the system learns which methods are performing better in different contexts. The metric for tracking user behaviour could be as simple as monitoring the number of accepted or rejected suggestions. This approach has the added benefit of gathering data about when different approaches work best, valuable information as predictive text entry reaches higher performance thresholds.

4.15 Phrase completion

The current focus of our commonsense system is word completion. This does not take full advantage of the semantic links that ConceptNet can provide among concepts. As demonstrated by Liu [26], commonsense knowledge is unique in its ability to understand context in language and semantic relationships among words. Commonsense knowledge is well-suited for phrase expansion, which would allow a predictive text entry system based on commonsense to effectively predict phrase completions.

4.16 Natural language processing

This first evaluation was meant to serve as a baseline comparison. As such, none of the conditions made use of language models or part of speech taggers. Clearly, these would have improved performance across all conditions. In designing future predictive typing aids, it would be worth exploring how different natural language processing techniques could further improve performance. If natural language processing was performed while users entered text messages, commonsense knowledge could be leveraged even more to improve word completion. For instance, given the phrase 'I'll meet you at the…' the system could know that the next word should be linked *isa→place* in ConceptNet, or 'I spoke to…' should be followed by a word that is linked *isa→person* in ConceptNet.

4.17 Speech recognition error correction

We are in the process of applying similar techniques to speech recognition systems [27]. This commonsense approach to predictive text entry can be used to improve error correction interfaces for such systems, as well as to disambiguate phonetically similar words and improve overall speech recognition accuracy.

5. Comparison of system architectures

Figure 9 displays the various ways the three applications described in this paper rely on our commonsense knowledge. All three applications use the context of a concept from ConceptNet. In addition, the social networking application uses analogous concepts from ConceptNet, and GloBuddy uses commonsense phrases taken directly out of the Open Mind corpus. Our word completion application accesses commonsense knowledge stored locally in memory to avoid network lag, but uses a subset of ConceptNet due to memory constraints. The other two applications access commonsense knowledge over the Internet using Web Services.

6. Related work

Because broad-spectrum application of commonsense reasoning is still not widespread, we cannot yet point to very many applications, especially regarding mobile devices, that use it. However, we do not want to leave the impression that we at MIT are alone in pursuing this. Lenat's Cycorp [28] currently has the largest commonsense knowledge base and the longest experience in creating applications with it. There are also many projects that take a different and more

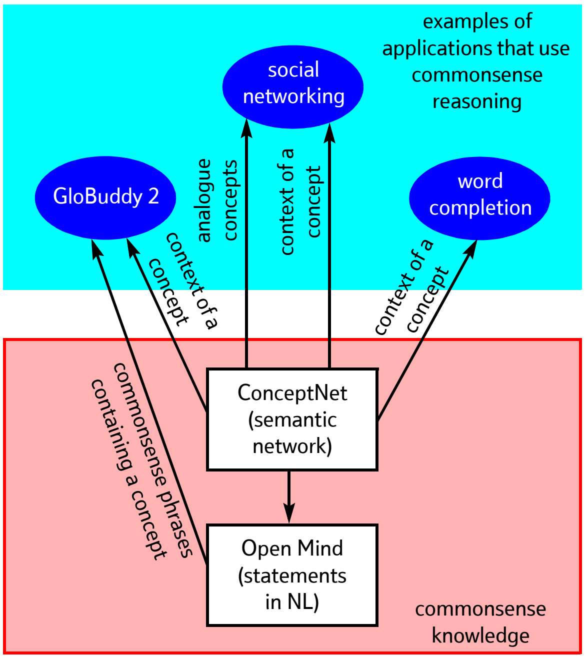

Fig 9 Comparison of system architectures.

formalistic slant to the commonsense problem, following the original idea of McCarthy [29]. A recent overview can be found in an article by Morgenstern and Davis [30]. Their approach is to isolate some particular capability of commonsense reasoning that they consider fundamental, and attempt to do an exhaustive axiomatic analysis of it. There are also several resources that provide broad-spectrum knowledge, but not what might be considered commonsense in the sense of contingent knowledge about everyday life. For example, WordNet [5] is a popular resource used in many AI natural language programs that provides word sense disambiguation, hypernyms and hyponyms. However, while WordNet [5] follows a more formal and taxonomic structure, ConceptNet [4] contains everyday knowledge about the world. The structure and creation of ConceptNet is described in Liu and Singh [4]. Additional information about research in commonsense reasoning at the MIT Media Lab can be found in Singh et al [14, 31].

7. Future work

The applications shown in this paper are only the first set of examples of how commonsense knowledge can be used to improve mobile computing. Commonsense knowledge could also be used in many other ways, including improving location-based services, and enabling mobile devices to query the Semantic Web [32].

7.1 Making better use of contextual information

In the future, commonsense knowledge can help mobile software applications make better use of contextual information like time, location, and personal data, allowing them to understand the user's context.

For instance, location-based information could improve all three of the applications discussed in this paper. GloBuddy could display pre-translated localised vocabularies of words based on where the user was physically located (in a museum, at the beach, etc). The social networking application could take user's locations into account when calculating search results (one friend is currently at a basketball court). And the word completion application could weight predictions based on contextual information like the user's location (anticipating the words 'flight' and 'delayed' when the user is located at an airport). Once a device knows the location of the user, it needs to know something about that location. The mobile device will need to know something about museums, beaches, basketball courts, airports, and thousands of other locations. Location-based information by itself is not enough to create intelligent mobile applications; they also need access to commonsense knowledge.

Future mobile devices should also make better use of a user's personal data. This personal data, along with commonsense knowledge could allow mobile devices to proactively provide their users with just-in-time information. For instance, if a user receives an e-mail saying that their brother will be flying in next Thursday, the user's mobile device could proactively:

- schedule an appointment to pick their brother up in the user's calendar,
- retrieve driving directions to the airport from the user's house ahead of time,
- track the flight in real time.

Creating a proactive application like this would require a mixture of personal data and commonsense knowledge. To complete this task, the application would need access to private information like its user's inbox, and home address, but it would also need to know about what tasks people commonly do when they pick someone up from the airport. Instead of being hard coded, this procedural information should come out of a corpus of commonsense knowledge so that the agent can proactively respond to thousands of situations. Commonsense knowledge architectures like LifeNet and StoryNet [13, 14], which model temporal and procedural information, could be used to create an application like this.

8. Conclusions — challenges and opportunities with mobile devices

Despite major differences in both the form factor and use of mobile devices, the vast majority of their software applications are simply smaller versions of the software applications commonly found in personal computers. The tendency to simply minimise PC-based software for mobile devices ignores one of their best attributes — people carry them everywhere. Because mobile devices are used in a much wider range of situations than a desk-bound computer, new opportunities emerge to proactively provide intelligent and appropriate assistance to the user in a just-in-time fashion. However, to appropriately respond to the user's current situation, mobile software applications need a better understanding of the world their users inhabit. GloBuddy 2 [6], our match making agent [7], and Eagle and Singh's research in understanding the topic of casual conversations [15], demonstrate some fundamental uses of commonsense knowledge. These applications use commonsense reasoning to understand their user's situations and goals, and take advantage of the wide variety of situations mobile devices are used in.

One of the biggest challenges facing mobile devices is dealing with their limited input and output. The only way to maintain functionality while minimising a software application's user interface is to make it more intelligent, to give it a better understanding of its user. Access to commonsense reasoning can reduce a mobile application's need for explicit user input because it can make better guesses about what the user might want. This is demonstrated by our text entry application [8, 9].

mobile applications need a better understanding of the world their users inhabit

By leveraging commonsense reasoning, mobile applications can both fulfill the opportunity of understanding their user by providing contextually relevant information in a just-in-time fashion, and they can overcome the challenges of their limited user interfaces.

Acknowledgments

The authors would like to thank Push Singh and Hugo Liu for their helpful feedback and for breaking new ground with Open Mind Common Sense and ConceptNet. Thanks also to Kevin Brooks and Angela Chang at the Motorola Advanced Concepts Group, and Paul Wisner and Franklin Reynolds at the Nokia Research Center for generously contributing their expertise and phones for our various implementations.

References

1 Open Mind Common Sense Project — http://commonsense.media.mit.edu/

2 Singh P: 'The Open Mind Common Sense Project', KurzweilAI.net (2002) — http://www.kurzweilai.net/meme/frame.html?main=/articles/art0371.html

3 Singh P, Lin T, Mueller E, Lim G Perkins T and Li Zhu Wan: 'Open Mind Common Sense: knowledge acquisition from the general public', Proc Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems (2002).

4 Liu H and Singh P: 'ConceptNet — a practical commonsense reasoning tool-kit', BT Technol J, 22, No 4, pp .... (October 2004).

5 Fellbaum C: 'WordNet: An Electronic Lexical Database', MIT Press, Cambridge, MA, USA (1998).

6 GloBuddy 2 — http://agents.media.mit.edu/projects/globuddy2/

7 Real time searches on a local social network — http://agents.media.mit.edu/projects/socialnetwork/

8 A commonsense approach to predictive text entry — http://agents.media.mit.edu/projects/textentry/

9 Stocky T, Faaborg A and Lieberman H: 'A commonsense approach to predictive text entry', Conference on Human Factors in Computing Systems (CHI 04), Vienna, Austria (2004).

10 Babel Fish Translation Services — http://babelfish.altavista.com/

11 Musa R, Scheidegger M, Kulas A and Anguilet Y: 'GloBuddy, a dynamic broad context phrase book', Proc Context 2003, Stanford, CA, USA, pp 467—474 (2003).

12 Berlitz: 'Spanish Phrase Book', Berlitz Publications Company (2001).

13 Singh P and Williams W: 'LifeNet: a propositional model of ordinary human activity', Submitted to the DC-KCAP (2003) — http://web.media.mit.edu/~push/LifeNet.pdf

14 Singh P, Barry B and Liu H: 'Teaching machines about everyday life', BT Technol J, 22, No 4, pp 227—240 (October 2004).

15 Eagle N, Singh P, Pentland A: 'Common sense conversations: understanding casual conversation using a common sense database', Proc AI2IA Workshop at IJCAI (2003).

16 Vivacqua A, and Lieberman H: 'Agents to assist in finding help', ACM Conference on Computers and Human Interface CHI (2000).

17 Friendster — http://www.friendster.com

18 Liu H, Lieberman H and Selker T: 'GOOSE: A goal-oriented search engine with commonsense', in De Bra P, Brusilovsky P, Conejo R (Eds): 'Adaptive hypermedia and adaptive web-based systems', Second International Conference, AH (2002).

19 Witten I H, Moffat A and Bell T C: 'Managing gigabytes: compressing and indexing documents and images', Morgan Kaufmann, San Francisco, CA (1999).

20 Conrad R and Longman D J A: 'Standard typewriter versus chord keyboard: an experimental comparison', Ergonomics, 8, pp 77—88 (1965).

21 Zhai S and Kristensson P O: 'Shorthand writing on stylus keyboard', Proc CHI 2003, CHI Letters, 5, No 1, pp 97—104 (2003).

22 Silfverberg M, MacKenzie I S and Korhonen P: 'Predicting text entry speed on mobile phones', Proc CHI 2000, CHI Letters, 2, No 1, pp 9—16 (2000).

23 Karat C-M, Halverson C, Horn D and Karat J: 'Patterns of entry and correction in large vocabulary continuous speech recognition systems', Proc CHI 2002, pp 568—575 (2002).

24 Darragh J J, Witten I H and James M L: 'The reactive keyboard: a predictive typing aid', IEEE Computer, 23, No 11, pp 41—49 (1990).

25 Zeno S et al: 'The Educator's Word Frequency Guide', Touchstone Applied Science Associates (1995).

26 Liu H: 'Unpacking meaning from words: a context-centered approach to computational lexicon design', Proc CONTEXT 2003, pp 218—232 (2003).

27 Using commonsense reasoning to improve speech recognition — http://agents.media.mit.edu/projects/voice/

28 Lenat D B: 'CYC: A large-scale investment in knowledge infrastructure', Communications of the ACM, 38, No 11, pp 33—38 (1995).

29 McCarthy J: 'Computers with Common Sense', in Lipschitz V (Ed): 'Formalisation of Common Sense', Ablex (1959).

30 Davis E and Morgenstern L: 'Progress in formal commonsense reasoning', Introduction to Special Issue on Formalisation of Common Sense, Artificial Intelligence Journal, 153, pp 1—12 (2004).

31 Singh P, Minsky M and Eslick I: 'Computing commonsense', BT Technol J, 22, No 4, pp 201—210 (October 2004).

32 Using commonsense reasoning to enable the Semantic Web — http://agents.media.mit.edu/projects/semanticweb/

Henry Lieberman's research focuses on the intersection of artificial intelligence and the human interface. He directs the Lab's Software Agents group, which creates intelligent software that helps users through interactive interfaces. In addition, he is working on agents for browsing the Web and for digital photography, and has built an interactive graphic editor that learns from examples and annotation on images and video. He worked with the late Muriel Cooper in developing systems that support intelligent visual design. Before joining the Media Lab he was a researcher at the MIT AI Lab, and worked with Seymour Papert on the educational language Logo. He holds a doctoral-equivalent degree (Habilitation) from the University of Paris VI and was a visiting professor there in 1989—90.

José Espinosa is a master's student in the Software Agents group at the Media Lab. He is interested in improving the interactions between people and computers, handhelds, and mobile phones.

He is trying to build applications with commonsense, in order to provide computers with understanding of the environments in which humans live. By exploring the differences in commonsense across persons and cultures, he hopes to improve the personalisation of user interfaces.

He received his undergraduate degree in computer engineering from the University of Mexico, where he also worked as a research assistant for two years.

Alexander Faaborg is a research assistant in the Media Lab's Software Agents group. His research focuses on improving software applications by giving them access to commonsense knowledge. This knowledge about human existence enables applications to act more intelligently, improving human computer interaction. Prototypes he has developed scale from improving word prediction on cell-phones to creating a next-generation Web browser. Faaborg graduated magna cum laude from Cornell University as a member of the College Scholar Program with a BA in information science and a concentration in cognitive science.

Tom Stocky researches commonsense computing, including an alternative approach to predictive text entry based on commonsense reasoning.

Mobile device implementations of this commonsense predictive typing aid have demonstrated the system in multiple computing environments.

He is currently a second-year master's student in the Media Lab's Software Agents group.

He earned his BS and MEng in electrical engineering and computer science from MIT.